



論文目録

報告番号	甲工第25号	氏名	松浦洋司
学位論文題目	超音波探傷試験評価のためのファジィエキスパートシステムの構築に関する研究		
論文の目次			
第1章 序論			
第2章 超音波探傷試験とエキスパートシステム			
2・1 超音波探傷試験			
2・2 エキスパートシステム			
2・3 非破壊試験を対象としたエキスパートシステム			
2・4 非破壊試験を対象としたエキスパートシステムの問題点			
第3章 超音波探傷試験のためのファジィエキスパートシステム			
3・1 ファジィエキスパートシステムの現状			
3・2 プロットタイプの構築			
第4章 ファジィ結合演算子の選定およびメンバシップ関数の改良			
4・1 はじめに			
4・2 検査技術者による経験的知識およびあいまいさの相違			
4・3 欠陥発生部位推定用メンバシップ関数			
4・4 本システムの構成と推論方法			
4・5 ファジィ結合演算子			
4・6 推論結果			
4・7 おわりに			
第5章 パソコン用システムの構築および妨害エコー判定用知識の補充			
5・1 はじめに			
5・2 概要			
5・3 推論時間の短縮			
5・4 妨害エコーの判定			
5・5 適用結果			
5・6 おわりに			
第6章 結論			
参考文献			
付録 超音波探傷試験評価のためのファジィエキスパートシステムのプログラム			
参考論文			
主論文			
(1) 超音波探傷試験評価のためのファジィエキスパートシステム (ファジィプロダクションルールの適用), 松浦洋司, 芳村敏夫, 川人明美 平成4年7月発行 日本機械学会論文集C編			
(2) 超音波探傷試験評価のためのファジィエキスパートシステム (ファジィ結合演算子の選定およびメンバシップ関数の改良), 松浦洋司, 芳村敏夫, 川人明美 平成4年11月発行 日本機械学会論文集C編			
(3) 超音波探傷試験評価のためのファジィエキスパートシステム (パーソナルコンピュータ用システムの構築および妨害エコー判定用知識の補充) 松浦洋司, 芳村敏夫, 森本和憲, 川人明美 平成5年10月発行 日本機械学会論文集C編			
副論文			
(1) A DIAGNOSTIC EXPERT SYSTEM FOR ULTRASONIC TESTING USING THE FUZZY REASONING, Hiroshi MATSUURA, Toshio YOSHIMURA, Akiyoshi KAWAHITO, Proceedings of the 2nd International Conference on Fuzzy Logic & Neural Networks, July 17-22, 1992			

備考

- 論文題目は、用語が英語以外の外国語のときは日本語訳をつけて、外国語、日本語の順に列記すること。
- 参考論文は、論文題目、著者名、公刊の方法及び時期を順に明記すること。
- 参考論文は、博士論文の場合に記載すること。

論文内容要旨

報告番号	甲工第25号	氏名	松浦洋司
学位論文題目	超音波探傷試験評価のためのファジィエキスパートシステムの構築に関する研究		
内容要旨			
<p>各種溶接構造物の溶接欠陥の検査として非破壊検査が行われており、その中で超音波探傷試験は作業性の良さや装置の性能の向上などにより、最近広く用いられるようになってきている。超音波探傷試験は、資格を持った検査技術者によって行われており、試験結果から欠陥の種類、欠陥寸法および欠陥等級を推定し、欠陥の可否を判定している。しかしながら、これらの推定にはかなりの熟練が必要とされるばかりでなく、推定規準にあいまいな点が多く検査技術者の主観の違いによる推定結果の差異が問題となることが少なくない。また、近年の労働力不足により検査技術者の育成が難しくなっている。</p> <p>一方、各分野において専門家の知識や経験を知識ベース化し、初心者でも専門家と同じ様に判定ができるエキスパートシステムの研究が行われ、一部実用化されている。最近では、専門家の知識や測定データなどに含まれるあいまいさをファジィ理論を用いて扱うファジィエキスパートシステムの研究も行われている。</p> <p>以上のような観点から、本研究では超音波探傷試験評価のためのファジィエキスパートシステムの構築に関する研究を行った。</p> <p>はじめに、上記のシステムのプロットタイプの構築を行った。システムはワークステーション上に、Common Lisp 言語を用いて開発したもので、推論エンジン、知識ベースおよびワーキングメモリから構成されている。通常、知識やデータの量が多くなると推論に要する時間が長くなるので、この問題を解消するために知識やデータをグループ化する黑板モデルを採用した。その際に、知識はIF-THEN 形式のプロダクションルールで表現し、フレームモデルによってグループ化した。知識は検査技術者から獲得し、その知識に含まれるあいまいさをメンバシップ関数や確信度を用いて表現した。推論にはあいまいさを扱うことができるようにファジィ推論を用いた。開発したシステムの推論結果と検査技術者による推定結果との比較から、本システムの推論精度の高さが確認された。しかし、本システムにおいて一部のデータに不適切な結果が得られていることとメンバシップ関数が一人の検査技術者の経験的知識に基づいていることおよび妨害エコーについての判定ができないことから、改良の余地が残されていることが指摘された。</p> <p>次に、ファジィ推論の中で問題となる結合演算子の選定およびメンバシップ関数の改良を行った。結合演算子は多数提案されているので、それらを本システムに適用し最も推論精度の良いものを選定した。メンバシップ関数については、複数の検査技術者によって差異があることから、それらの差異を調整しチューニングを行い作成した。これによって、プロットタイプでみられた不適切な推論結果は解消され、精度が向上したことが確認された。</p> <p>さらに、これまでのシステムは、ワークステーション上に構築されているため持ち運びが困難なことおよび妨害エコーについて考慮されていないことなどの問題があった。そこで、持ち運びが可能なパソコン用のシステムの構築および妨害エコーについての知識の補充を行った。パソコン用のシステムの構築の際には推論時間の短縮についても検討した。妨害エコー判定用の知識に含まれるあいまいさについてはこれまでと同様にメンバシップ関数等を用いて扱った。検査技術者の推定結果との比較より、妨害エコーについても精度良く推論できることが確認された。</p> <p>本研究によって超音波探傷試験の結果から欠陥の状況を精度良く推論できるファジィエキスパートシステムが構築できた。さらに、本研究で用いたファジィエキスパートシステムの開発手法は、他のシステムの開発に十分適用できるものと思われる。</p>			



超音波探傷試験評価のためのファジィ  
エキスパートシステムの構築に関する研究

1994年3月

松 浦 洋 司

②

超音波探傷試験評価のためのファジィ  
エキスパートシステムの構築に関する研究

1994年3月

松 浦 洋 司



## 目 次

第1章 序 論	1
第2章 超音波探傷試験とエキスパートシステム	
2・1 超音波探傷試験	4
2・1・1 非破壊試験	4
2・1・2 超音波探傷試験の種類	8
2・1・3 超音波探傷試験の適用分野	13
2・1・4 欠陥の評価方法	16
2・2 エキスパートシステム	20
2・2・1 人工知能とエキスパートシステム	20
2・2・2 知識工学とエキスパートシステム	21
2・2・3 知識表現モデル	23
2・2・4 開発されたエキスパートシステムの特徴	24
2・3 非破壊試験を対象としたエキスパートシステム	26
2・3・1 超音波探傷試験による溶接欠陥の 検出と評価のためのエキスパートシステム	26
2・3・2 放射線透過写真の自動合否判定システムの開発	27
2・3・3 超音波探傷試験結果評価支援エキスパートシステム	28
2・4 非破壊試験を対象としたエキスパートシステムの問題点	28
第3章 超音波探傷試験評価のためのファジィエキスパートシステム	
3・1 ファジィエキスパートシステムの現状	30
3・1・1 ファジィ理論	30
3・1・2 ファジィエキスパートシステム	36
3・2 プロットタイプの構築	40
3・2・1 はじめに	40
3・2・2 超音波探傷試験評価とその問題点	41
3・2・3 メンバシップ関数および確信度	43
(1) 欠陥発生部位推定用のメンバシップ関数	
(2) データ判定用のメンバシップ関数	
(3) 確信度	
3・2・4 システムの構成と手順	47
(1) 概要	
(2) 推論方法	
3・2・5 適用結果	51
3・2・6 おわりに	58



第4章 ファジィ結合演算子の選定およびメンバシップ関数の改良	
4・1 はじめに	59
4・2 検査技術者による経験的知識およびあいまいさの相違	59
4・3 欠陥発生部位推定用のメンバシップ関数	61
4・4 推論方法	62
4・5 ファジィ結合演算子	64
4・5・1 種々のファジィ結合演算子	64
4・5・2 本システムで用いたファジィ結合演算子	65
4・6 推論結果	68
4・6・1 ファジィ結合演算子の選定	68
4・6・2 本システムと検査技術者による推定結果の比較	73
4・7 おわりに	75
第5章 パソコン用システムの構築および妨害エコー判定用知識の補充	
5・1 はじめに	76
5・2 概要	76
5・2・1 パソコン用システム	76
5・2・2 妨害エコー	77
5・3 推論時間の短縮	78
5・3・1 黒板モデルの改良	78
5・3・2 推論時間の変化	78
5・4 妨害エコーの判定	80
5・4・1 妨害エコー	80
5・4・2 妨害エコー判定用の知識	80
5・4・3 知識ベース	87
5・5 適用結果	87
5・6 おわりに	92
第6章 結 論	93
参考文献	95
謝 辞	101
付録 超音波探傷試験評価のためのファジィエキスパートシステムのプログラム	

## 第1章 序 論

溶接は工業製品の生産工程の中で最も重要な加工技術の一つであり、溶接とこれに関係する技術は、多数の研究者や技術者の努力により近年著しい発展を遂げている。現在では、特殊な材料を除けば比較的簡便にこれらの技術を製品の生産工程の中に取り入れて利用できるようなっている。このため、溶接技術は各種構造物・船舶・自動車・大小機械製品・家電製品・家庭用品などを始め、ほとんどあらゆる金属製品の製作に広く利用され、生産性の向上に寄与している。しかしながら、溶接部を起点とする製品の破壊や破損事例が大小を問わずしばしば発生していることも事実である。したがって、溶接が広く利用されるにつれて、当然ながら溶接技術に対して高い信頼性が要求されてくるといえる。

溶接が健全であるかどうかを調べる方法として非破壊試験があり、その信頼性は、近年これまでの試験データの蓄積などから著しく向上している。非破壊試験の種類としては多数開発されているが、溶接内部の欠陥の検出には特性上超音波探傷試験および放射線透過試験が適している<sup>(1)</sup>。

一般に、超音波探傷試験より放射線透過試験の方が欠陥の判定は容易であるとされているが、大型構造物のように放射線による写真の撮影が困難なものに対しては、超音波探傷試験がよく用いられる。特に、現場での作業が要求される鋼構造建築物については、溶接部の非破壊検査は日本建築学会「鋼構造建築溶接部の超音波探傷検査基準・同解説」<sup>(2)</sup>に基づく超音波探傷試験であることが規定されている。しかしながら、超音波探傷試験の結果から溶接欠陥の評価をするにはかなりの熟練が必要とされ、その実施は超音波探傷検査技術者の資格を持った者に限られている。また、ボイラーおよび圧力容器などについて



は、溶接部における非破壊検査として放射線透過試験を実施することが規格により定まっている。しかし、放射線透過試験が困難な部分については超音波探傷試験を代用することが一部で認められている。

一方、各分野において専門家の知識や経験を知識ベース化し、初心者でも問題が解決できるようなエキスパートシステムについての研究が盛んに行われており、特に特殊な専門的知識が必要とされる問題については完成度の高いシステムが開発され、一部実用化され始めている<sup>(3)</sup>。このようなシステムは、近年の労働力不足により専門家の育成が困難となっていることと相まって今後ますます要求度が高くなっていくものと考えられる。超音波探傷試験についても、その結果を評価するにはかなりの熟練が必要なことと近年の労働力不足による検査技術者の不足などから試験結果を評価するシステムの開発が望まれている。

そこで本論文では、非破壊試験としてその需要が非常に多い超音波探傷試験について、その結果を評価するためのエキスパートシステムの構築を目的とする。その際に、検査技術者の持つ経験的知識や測定データに含まれているあいまいさおよび不確実さについてファジィ理論を適用する。また、構築したシステムを実際の試験データに適用し、その有効性について検討する。

第2章では、非破壊試験およびエキスパートシステムについて説明する。非破壊試験としては、溶接内部の欠陥の検査に適している超音波探傷試験と放射線透過試験についてその特徴を説明し、さらに超音波探傷試験について原理、適用分野および評価方法を説明する。エキスパートシステムについては過去の研究内容、開発されたエキスパートシステム<sup>(4)-(10)</sup>の特徴などについて、また非破壊試験を対象としたエキスパートシステムの開発例<sup>(11)-(17)</sup>およびそれらの問題点について説明する。

第3章では、ファジィ理論を用いたエキスパートシステムおよび開

発した超音波探傷試験評価のためのファジィエキスパートシステムのプロットタイプ<sup>(18)</sup>について説明する。最初に、あいまいさや不確実さを表現するためのファジィ理論およびファジィ理論を用いた推論方法について説明し、その後、開発したプロットタイプについて、まず超音波探傷試験の結果からの欠陥の評価方法および用いたメンバシップ関数と確信度について、次にシステムの構成およびファジィ推論の手順を、最後にシステムの適用結果について説明する。

第4章では、第3章で開発したプロットタイプの適用結果の一部において不適当な結果が得られたことに対する改良結果<sup>(19)</sup>を述べる。不適当な結果が得られた主な理由としては、ファジィ推論に用いた結合演算子およびメンバシップ関数の形状が適切でなかったことが考えられる。システムに最適な結合演算子は対象とする問題によって異なるため、その選定には十分な検討が必要である。また、プロットタイプではメンバシップ関数の形状を一人の専門家の経験的知識によってのみ決めたが、経験的知識や主観が片寄る可能性があると予想される。そこで、ファジィ結合演算子およびメンバシップ関数をいろいろと変更し、そのときのシステムと専門家による推定結果とを比較することにより最適なファジィ結合演算子およびメンバシップ関数の形状を決定する。

第5章では、第3章および第4章で開発したシステムにおいて考慮されていなかった妨害エコーについても推定できるように知識を補充すると共に、より実用的なものにするためにシステムをパソコン上に移植する<sup>(20)</sup>。パソコン上のシステムとする際、推論時間が長くなることが問題となることから、推論時間の短縮方法についても検討する。

第6章では、本論文のまとめと今後の課題について述べる。



## 第2章 超音波探傷試験とエキスパートシステム

### 2・1 超音波探傷試験

#### 2・1・1 非破壊試験

材料・機器・構造物などの内部に欠陥が含まれているかどうかを調べるためには、これらを破断してその破面を肉眼で観察するのが最も確実な方法である。同一形状・寸法で同一の加工条件で施工した多量の製品が生産される場合には、その一群の製品の中から一部を抜き取って破壊試験を行うことにより、全数の製品の状況を推定することができる。しかしながら、破壊試験を行うとその材料・機器・構造物は実際に使用不可能となるため、注文生産のため同じ形状の製品が無い場合や小量生産の場合などには破壊試験を適用することが困難である。したがって、そのような場合には破壊試験の代わりに非破壊試験が用いられる。

非破壊試験とは対象物を破壊しないで、それらの性質、状態、内部構造、内部欠陥などを調べる方法である。その方法には、外観試験の他に表2・1<sup>(21)</sup>に示すような種々の非破壊試験がある。これらの方法は、材料が有している性質（例えば光、放射能、音波、超音波、電気、磁気などに対する物理的性質）が、組織の異常や欠陥の存在によって変化することを利用している。また、これらの非破壊試験の適用例として、材料および溶接部の欠陥検査、内部構造または内容物の調査、材料や機器の計測検査、材質調査、表面処理層の厚さ測定、ひずみ測定などがある。特に欠陥検査としての適用は最近の30年ほどの間に急速に普及し、製造時だけでなく、定期点検時もしくは稼動時にも適用されるようになっている。

構造物の溶接部は、欠陥検査としての非破壊試験が最もよく適用される部分と言える。溶接は局所的な鋳造操作であるので、急熱・急冷

表2・1 各種非破壊試験法

試験法の種類	試験法の摘要	試験の目的
浸透探傷試験	溶接部表面に特殊な浸透液を塗布して表面欠陥に浸透せしめ、微小欠陥も明らかに視認させる。	表面欠陥の位置、大きさ、形状の検出
磁粉探傷試験	被検体を磁化し、これに磁性粉末を散布することによって、欠陥部に現れる漏れ磁束を検知し欠陥の存在を検出する。	磁性体に対する表面欠陥および浅い内部欠陥の位置、大きさの検出
渦流探傷試験	交流のコイルを溶接部に当て、導体に誘導される渦電流の変化を捉えて欠陥の検出を行う。	表面欠陥および浅い内部欠陥の位置の検出
超音波探傷試験	超音波のインパルスを被検体の一面から入射させ、内部欠陥からの反射波の状況をブラウン管上で観察して欠陥の検出を行う。	内部欠陥の位置の検出
放射線透過試験	被検体にX線またはγ線を透過させ、内部欠陥の状況を写真フィルムに感光させて撮影し検出する。	内部および表面欠陥の位置、大きさ、形状の検出
AE（アコースティック・エミッション）試験	内部欠陥の成長中に放出される弾性エネルギーの音波を連続的に受信し、欠陥の存在およびその成長の状況を検知する。	内部欠陥の検知および破壊の予知、安全性の確認



による変質や各種の欠陥などの発生が考えられる。溶接部に発生する欠陥にはビード割れ，熱影響部割れ，溶け込み不良，融合不良，スラグ巻き込み，ブローホール，ピットあるいはアンダーカットなどがある。これらの欠陥のうち外観調査によって発見できるものはピットおよびアンダーカットもしくは表面に現われた割れで，ほとんどの欠陥は外観調査のみでは発見することは困難である。そこで，非破壊試験を行うことにより，溶接部の欠陥の有無を調査し溶接の良否の確認が行われている。溶接部の欠陥には内部に発生する欠陥の方が多いことから，溶接部の非破壊試験には内部の欠陥の検出に適している放射線透過試験および超音波探傷試験がよく用いられる。両者の試験の特徴を比較したものを表2・2<sup>(22)</sup>に示す。表に示す記号は絶対的なものではなく，傾向としてこのような場合が多いことを示している。以下に両者の特徴を簡単に説明する。

放射線透過試験はX線， $\gamma$ 線のような電磁波もしくは電子線， $\beta$ 線，中性子線などの粒子線を用い，物体を透過撮影し内部の状況を調べようとするものである。この試験では欠陥像がフィルム上に二次元的に得られるため，欠陥の有無や欠陥形状が比較的簡単に判定できることおよび試験結果の保存ができるといった利点がある。また，立体的な欠陥を検出しやすいといった特徴がある。しかし，その反面，欠陥の深さ方向の位置の推定や開口幅の小さい割れ，放射線の方法と角度をなす割れもしくは溶け込み不良の検出が困難なこと，設備が高価である，試験に時間がかかる，安全管理を十分行わなければいけないといった欠点がある。

超音波探傷試験は音波を材料の一端から送り，欠陥または端面からの反射波を捕らえ，その様子から欠陥に関する情報を得ようとするものである。反射波の表示方式としては，装置が簡単なAスコープと呼ばれるブラウン管(CRT)上の図形として表す方法が最もよく用いられている。超音波探傷試験は割れ，融合不良，溶け込み不良などの

表2・2 放射線透過試験と超音波探傷試験の比較

試験方法		超音波探傷法	放射線透過試験(直接撮影法)
原理	原理的方法	パルス反射法	透過法
	物理的エネルギー	弾性波	電磁波
	欠陥部における現象	健全部では反射波が生じないが，欠陥部では反射波が生じる。その程度は健全部と欠陥部の材質に係る。	健全部と欠陥部とは透過線量が異なる。その程度は健全部と欠陥部の材質および放射線の透過する方向の欠陥寸法に係る。
	情報を表示する機材	ブラウン管	X線フィルム
	表示の内容	欠陥部のエコー(反射波)が現れる。	健全部と欠陥部とは写真濃度が異なる。
理	検出しやすい欠陥の方向	超音波の進行方向に垂直な方向	放射線の進行方向に平行な方向
	検出しやすい欠陥の形状	超音波の進行方向に垂直に広がりのあるもの	放射線の進行方向に奥行きのあるもの
加工方法	鑄造材	○	◎
	鍛造材	◎	×
	圧延材	◎	×
	溶接部	○	◎
欠陥の検出	二枚割れ	◎	×
	ひき果	○	◎
	パイプ(鑄造材)	○	◎
	ブローホール	△	◎
	パイプ(溶接部)	△	◎
	溶け込み不良	○	○
	融合不良	○	△
	割れ	△	△
試験方法の特徴	スラグ巻き込み	○	◎
	欠陥の種類判別	△	◎
	欠陥の形状判別	△	◎
	欠陥の寸法判別	△	○
	欠陥の深さ方向の位置判別	◎	△
	試験結果記録の優劣	△	◎
	現場に判定者の立会いが不要	△	◎
	片面からのみの試験が可能	◎	×
	試験ができる肉厚の上限	◎	○
	試験ができる肉厚の下限	△	◎
	装置の小型軽量	◎	×
	試験の迅速性	◎	×
微	消耗品費が少額	◎	×
	総合費用が少額	◎	×
	安全管理	◎	△

◎非常によい ○よい △条件付きでよい ×よくない



検出に適していることやAスコープを用いる場合装置が安価であるといった利点がある。しかしその反面、ブローホールのような球状の欠陥や小さい欠陥を検出しにくいことや探傷結果の保存性が少ないといった欠点がある。さらに、良く用いられるAスコープでは情報量が少なく欠陥の判定が難しいため熟練が要求されるという問題がある。

以上のことから、放射線透過試験は小規模の溶接部（ボイラーおよび圧力容器などの溶接部）についてよく用いられ、規模の大きい構造物（原子力発電用圧力容器、建築構造物など）などには写真撮影が困難なことおよび試験に十分な時間がかけられないことなどから超音波探傷試験がよく用いられている。

#### 2・1・2 超音波探傷試験の種類<sup>(1)</sup>

人間の耳に聞こえる音波の周波数は20Hz～20kHzであり、これよりも高い周波数の音波を超音波と呼んでいる。このうち探傷に用いられるのは200kHz～25MHzの範囲のものが大部分であり、特に1～5MHzのものがよく用いられる。音波および超音波の本質は、空気、水、鋼など音の媒質における分子あるいは原子の機械的な振動であり、その機械的振動の伝搬が音波および超音波の伝搬である。固体内を伝搬する音波には縦波、横波および表面波の3つの波動が存在する。このうち探傷には縦波がよく用いられる。音波の伝搬する速度は周波数には関係なく、物質の種類によって定まった値を持っている。

超音波には以下の3つの重要な性質がある。

- (a) 指向性を持っており、特定の方向に集中して出すことができる。これは周波数が高い（すなわち波長が短い）ためであり、周波数が高いほど指向性が高くなる。
- (b) 材料内部に異物や欠陥あるいは接合面がある場合にはその境界面において反射する。この性質を利用して欠陥の検出を行うわけである。

- (c) 材料内部の欠陥などによって超音波が反射散乱される度合は、その欠陥の大きさと超音波の波長の関係によって決まる。欠陥が超音波の波長より大きい場合には反射散乱がよく行われ、逆に小さい場合には反射散乱はほとんど行われず欠陥の存在に関係なく通り過ぎる。

超音波探傷法は表2・3に示すように分類される。超音波を入射し、欠陥によって音波が反射することを利用するパルス反射法、欠陥によって透過する音波が散乱減少することを利用する透過法、共振による定常波の変化を利用する共振法がある。現在最も広く使用されている方法はパルス反射法である。パルス反射法には、超音波を送る送信探触子と反射してきた超音波を受ける受信探触子があり、両者を一つで共有する一探触子法と送・受信に分けて行う二探触子法がある。また、超音波を探傷面に垂直に入射させる垂直探傷法と探触面に対してある角度を持った方向に入射させる斜角探傷法がある。垂直探傷法に用いる垂直探触子および斜角探傷法に用いる斜角探触子を図2・1に示す。どちらも振動子からは縦波が発振される。垂直探触子では発振された縦波がそのまま被検査物に伝搬されるが、斜角探触子では振動子で発振された縦波は境界面で反射し被検査物には横波のみが伝搬することになる。どの検査法を適用し周波数をいくにするかは、検査対象によって経験的に判明しており、溶接部の探傷では特別な場合を除き斜角一探触子法がよく用いられる。

図2・2にパルス反射法の探傷方法を示す。これは探触子として垂直探触子を用いた場合のものである。探触子から発信された超音波は内部の欠陥や材料の底面から反射する。それらの反射波を表示したものを図2・3に示す。この表示方式は、Aスコープと呼ばれる方法で、装置が簡単なことから最もよく用いられている。縦軸は受信音圧に比例するものを表し、横軸は時間軸であり超音波が反射してくるまでの時間を示している。超音波が反射してくるまでの時間は超音波の発信

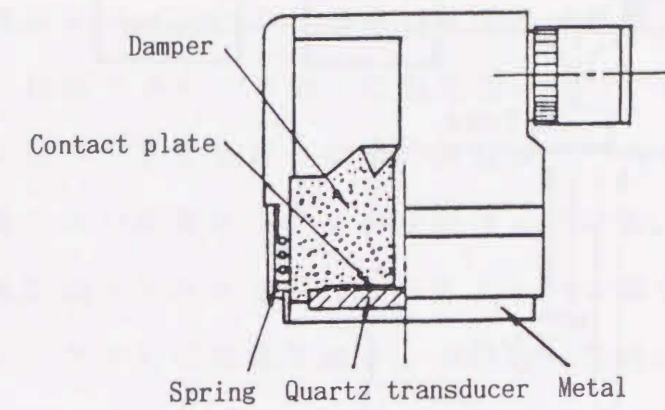


表2・3 超音波探傷法の種類と主な用途

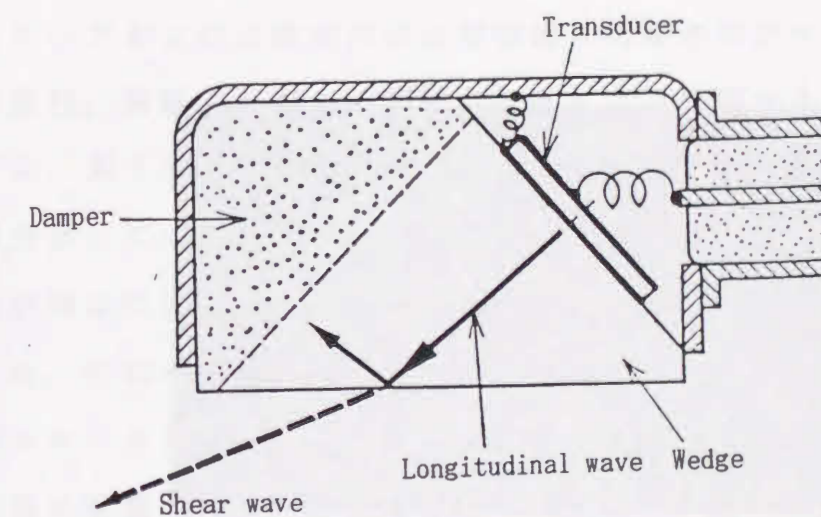
```

graph LR
    A[パルス反射法] --- B[一探触子法  
(直接法)  
(水浸法)]
    A --- C[二探触子法  
(直接法)]
    B --- D[垂直法 . . . . . 内部欠陥, 材質, 厚さ (最も一般的に使用される, 縦波使用)]
    B --- E[斜角法 . . . . . 溶接部, 管, ラミネーション, 表面近傍のきず (横波使用)]
    B --- F[表面波法 . . . . . 表面欠陥]
    B --- G[板波法 . . . . . 板, 管, 板の接着良否]
    C --- H[垂直法 . . . . . 表面近傍のきず]
    C --- I[斜角法 . . . . . 溶接部]
    C --- J[板波法 . . . . . 板]
    K[透過法] --- L[パルス法 . . . . . 鋳物の巣, 核燃料棒, 板  
(水浸法)]
    K --- M[連続波法]
    M --- N[普通の透過法 . . 板, 溶接部, 棒  
(直接法)  
(水浸法)]
    M --- O[共振透過法 . . . 板  
(水浸法)]
    P[共振法] --- Q[ブラウン管式 . . . . . 厚さ測定, 腐食度測定, 板のラミネーション  
(直接法)  
(水浸法)]
    P --- R[音響式 . . . . . 厚さ測定, 減衰によるきず, 材質  
(直接法)]

```



(a) 垂直探触子



(b) 斜角探触子

図2・1 探触子の構造



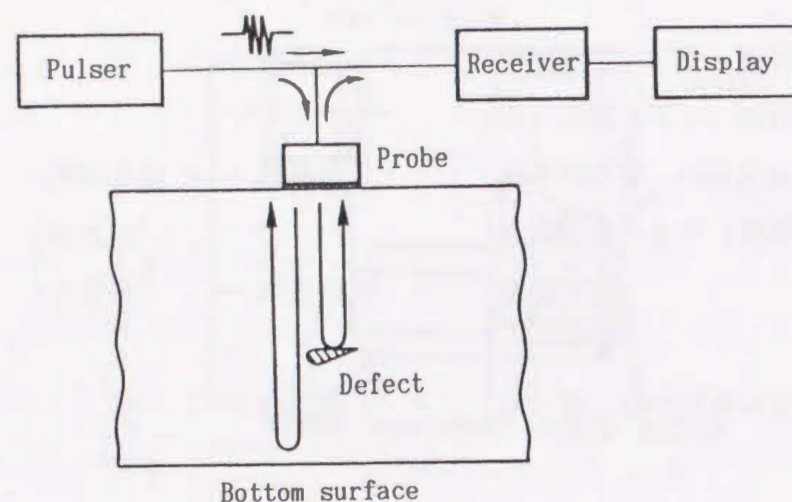


図 2・2 パルス反射法の原理

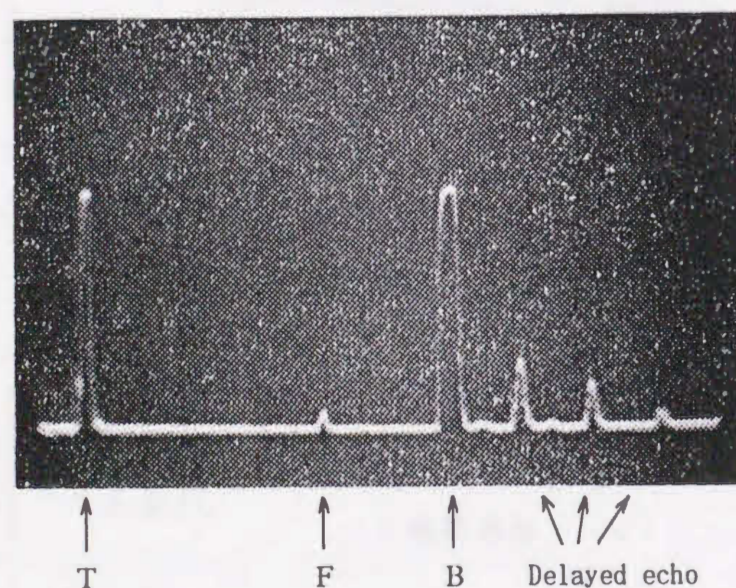


図 2・3 A スコープの表示例

点から反射源までの距離（ビーム路程）に比例するので、時間軸は距離軸とも呼ばれる。図 2・3 の T は送信パルス、F は欠陥エコー、B は底面エコーである。送信パルスとは探触子と被検査物との接触面からの反射であり、欠陥とは関係ないものであるが、距離軸のゼロ点調整を行うときに必要となる。欠陥エコーおよび底面エコーはそれぞれ欠陥および底面からの反射波を表している。

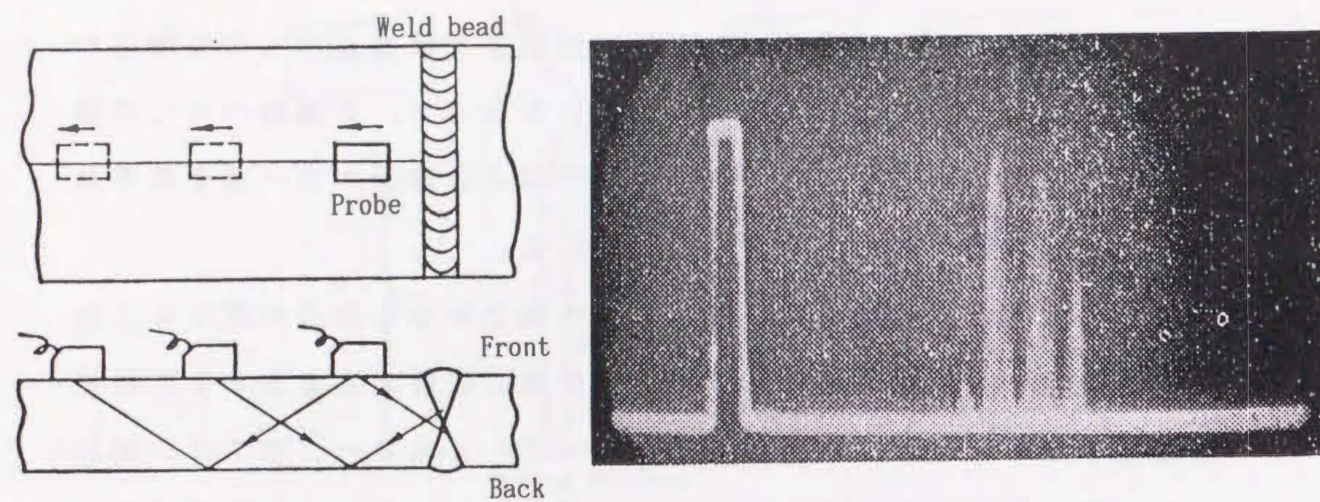
欠陥の大きさや形状によってエコーの高さやその波形が異なることから、エコーの位置や高さ、波形から欠陥の位置や大きさおよび形状を推定することが可能である。したがって、A スコープのエコー図形から溶接欠陥の有無を調べ、欠陥の判定を行うことにより溶接の良否を判定することができるが、情報量が少ないのでそれらの推定には熟練が必要となる。欠陥などによる反射波の表示方法には A スコープ以外に A スコープを重ね合わせた M A スコープ（図 2・4）や B スコープ（図 2・5）、X 線透過写真のような二次元図形に表す C スコープ（図 2・6）および立体表示法（図 2・7）などがある。B スコープ、C スコープおよび立体表示法は直感的に欠陥の形状や大きさが判断できる反面、装置が大型かつ高価になるという問題がある。このような方式は、原子炉容器などの自動超音波探傷試験装置の開発に利用され成果を挙げている<sup>(23) - (25)</sup>。

超音波は欠陥だけではなく、溶接の裏波や裏当金の角などからも反射する。これらによって得られたエコーは妨害エコーと呼ばれ、この妨害エコーと欠陥によるエコーを正確に判別することが超音波探傷を行う場合重要となる。超音波探傷ではブラウン管に現れた波形のみから欠陥の判定を行うと同時に妨害エコーの判別も行わなければならないため、非常に熟練が要求されるわけである。

### 2・1・3 超音波探傷試験の適用分野

超音波探傷試験は各種鉄鋼製品（厚鋼板、薄鋼板、ビレット、レールなど）をはじめ、航空機、原子力、自動車、建設機械、鉄道、鉄骨

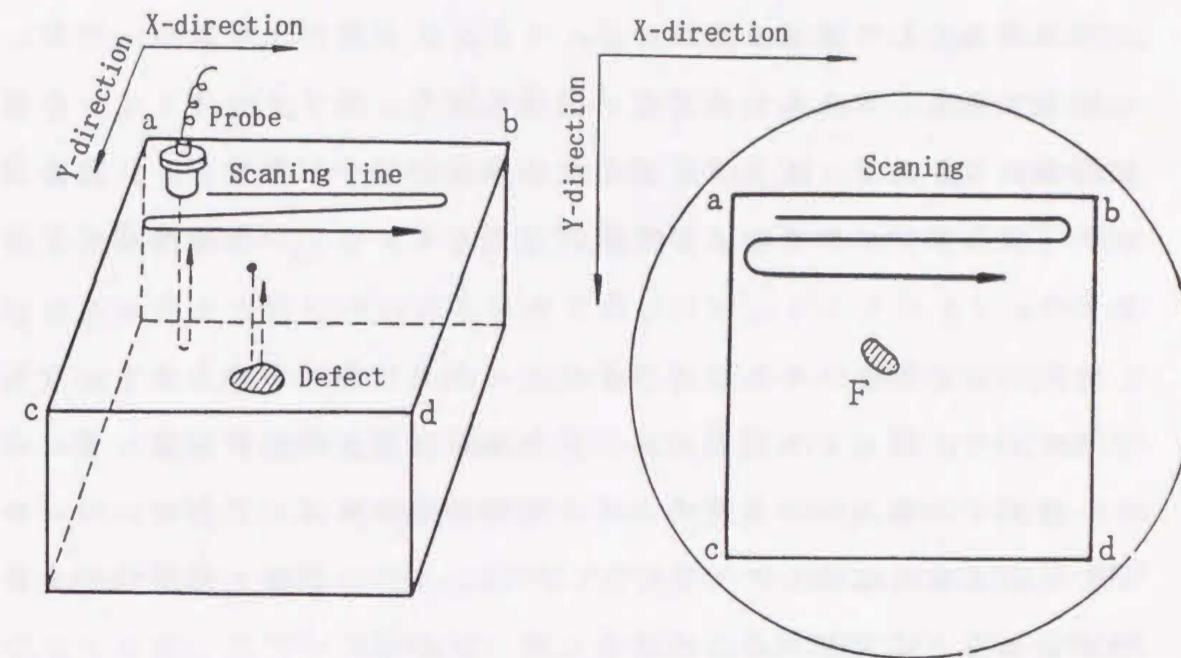




(a) 探傷方法

(b) 表示

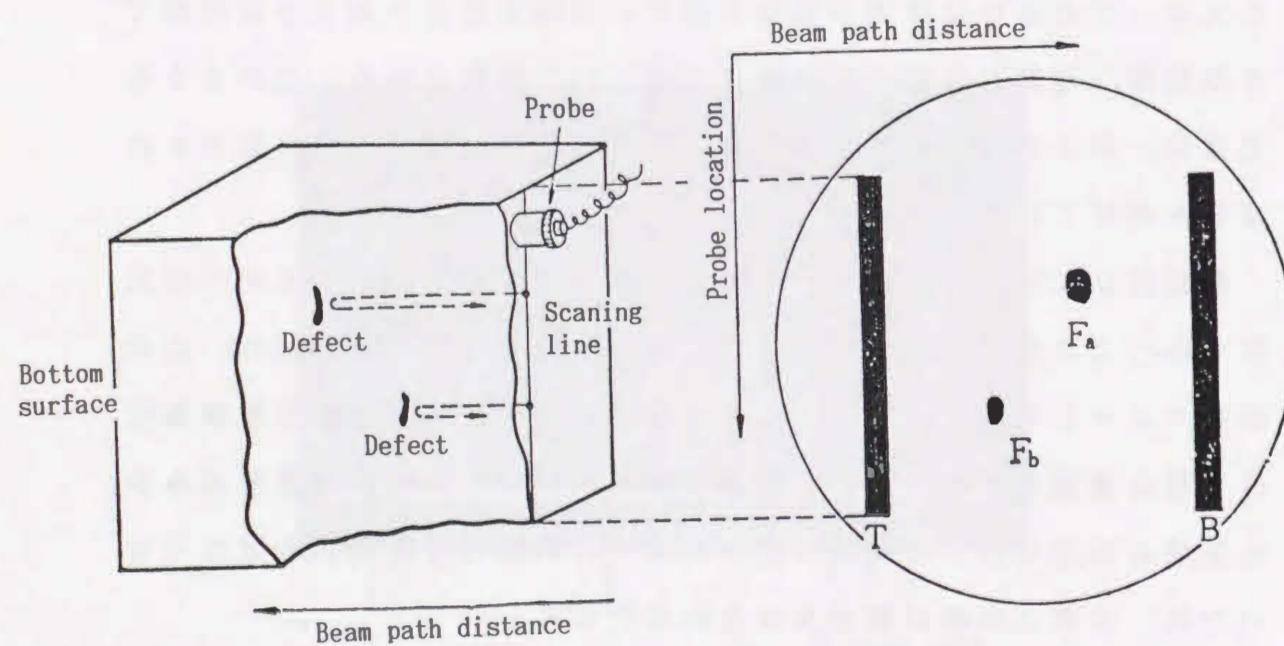
図 2・4 MA スコープの表示例



(a) 探傷方法

(b) 表示

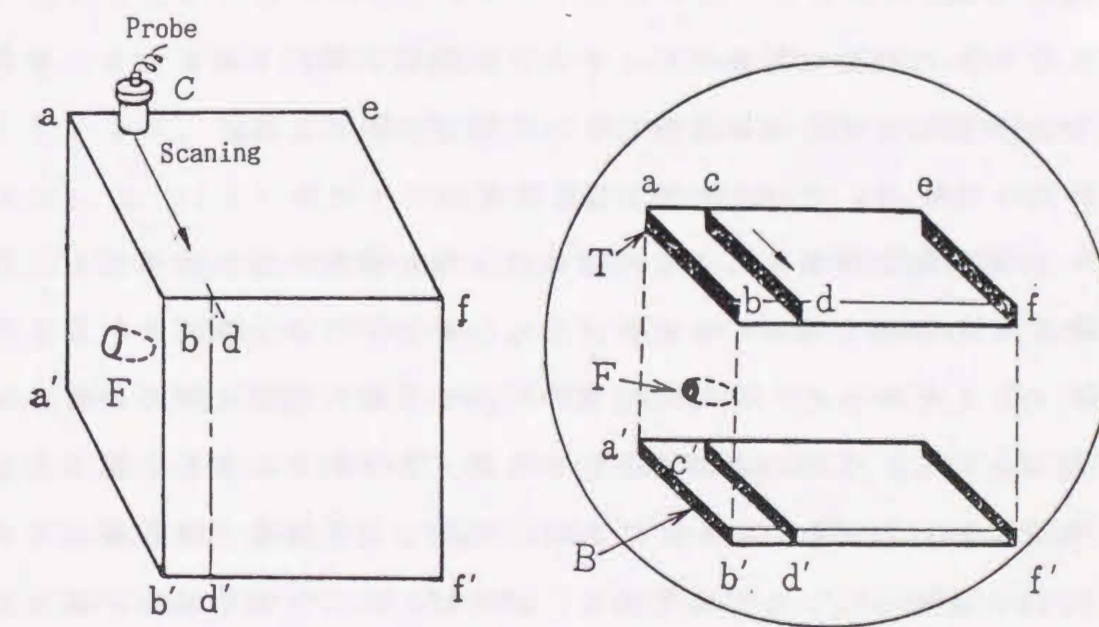
図 2・6 C スコープの表示例



(a) 探傷方法

(b) 表示

図 2・5 B スコープの表示例



(a) 探傷方法

(b) 表示

図 2・7 立体的表示法の表示例



構造物、圧力容器、石油パイプラインなどのいろいろな分野における非破壊検査として適用されている。また、生産性向上のため、全数、全領域保障を行うための省力化、高速処理化、多チャンネル化した自動探傷大型システムの要求と製品の出荷時検査や現場検査、自動探傷が不可能な部分の検査および経済的な面などからの小型探傷器の要求がある。

自動探傷大型システムは原子炉などの検査に用いられるようになってきており、最近では製造時の材料欠陥や溶接欠陥の評価だけでなく、使用中の構造物の材料の劣化や損傷の進展状況の把握のためにも用いられるようになっている<sup>(26) - (28)</sup>。また、種々の構造物の余寿命の診断などにも利用されるようになっている。

小型探傷器は建築物の溶接検査において使用頻度が高く、小型、軽量、高性能、AC電源不用などが要求される。そのため、反射波の表示法はAスコープが用いられる。したがって、ブラウン管上の波形のみから欠陥の判定を行わなければいけないため、長い経験と熟練が必要となる。

これらの検査は対象物毎にそれぞれ規格が設定されており、検査方法や欠陥の評価方法が規格によって規定されている。

#### 2・1・4 欠陥の評価方法

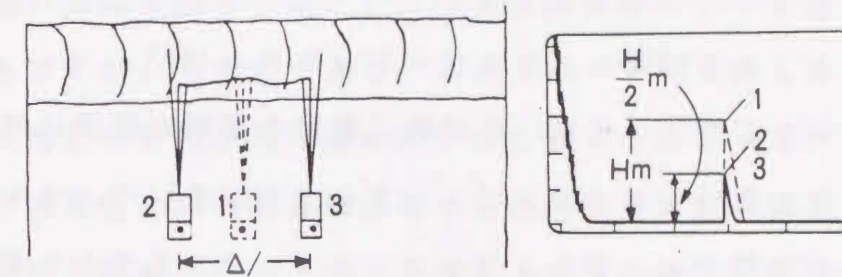
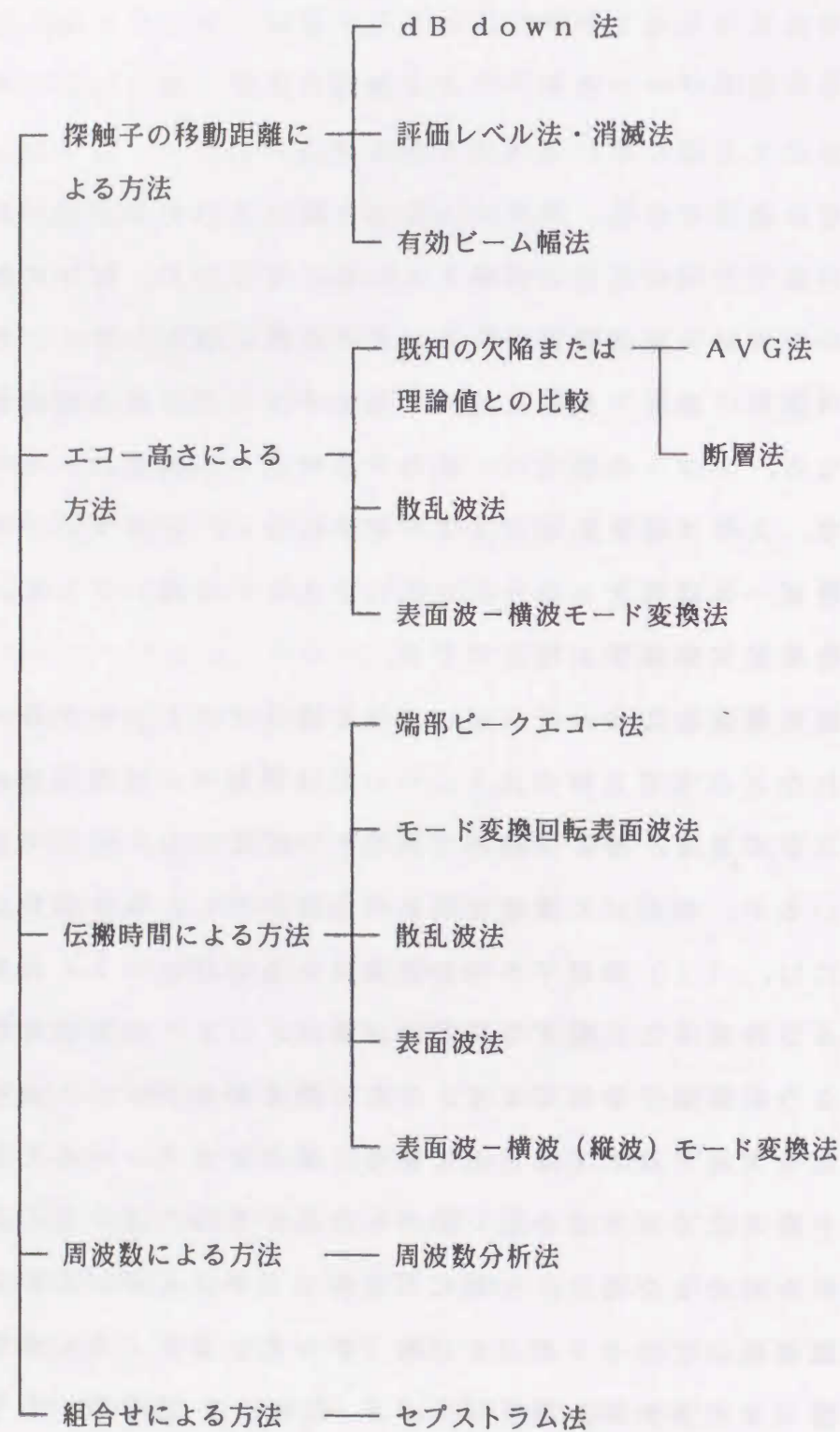
溶接欠陥を評価することの意味は欠陥の種類や寸法を推定し、最終的にその溶接を補修しなければならないかどうかの判定を行うことである。したがって、欠陥の種類や寸法が正確に推定できるかどうかが必要となる。欠陥の種類を推定する際、まず初めに欠陥の発生部位を推定しなければならない。これは、欠陥の発生部位が欠陥種類によってほぼ決まっているためである。例えば、ルート溶け込み不良はルート部に、融合不良は開先面もしくは溶接金属部に発生する。探触子から欠陥までの距離はブラウン管上のエコーの位置に比例しているため、探触子の位置がわかれば欠陥が溶接のどの部分（溶接金属部、ルート

部、開先面、熱影響部もしくは母材部）にあるかが推定できる。したがって、探触子の位置を求めるための基準線が必要である。この基準線はV、X形開先では中心線、レ、K形開先では開先の設けていない材の開先面とすることが多い。欠陥発生部位を推定した後、その推定結果とその他のデータを用いて欠陥種類を推定することになる。欠陥種類を推定する際に用いる欠陥発生部位以外のデータは主にエコーの高さとその波形である。エコーの高さは反射された超音波の量に比例し、その量は欠陥の形状に関係している。すなわち、球状の欠陥（ブローホールなど）は超音波がいろいろな角度に散乱しやすいため面状の欠陥（割れ、融合不良）に比べて探触子で受信される超音波の量が少なくなる。エコーの波形は、割れや溶け込み不良ではシャープに立ち上がり、スラグ巻き込みではギザギザになり、密集ブローホールでは色々なビーム路程でエコーが立ち上がるという違いがある。これらの情報を基に欠陥種類が推定できる。

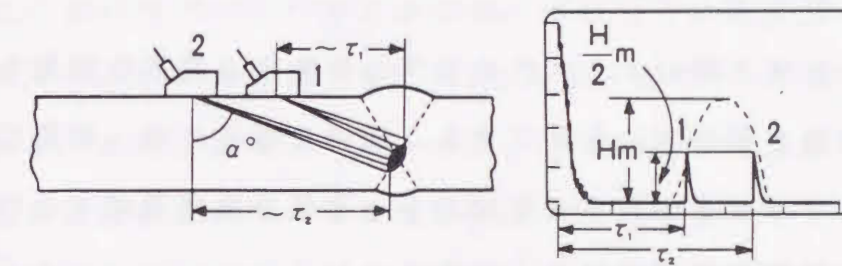
超音波探傷試験によって欠陥の寸法を推定する方法を表2・4に示す。割れなどの水平方向の長さについては探触子の移動距離から推定することができる。深さ方向の寸法についてもいろいろな方法が提案されているが、精度良く推定できる方法は少ない。深さ方向の寸法の測定法には、（1）探触子の移動距離による方法と（2）既知の欠陥寸法のエコー高さと比較する方法とがある。（1）の方法は図2・8に示すように探触子を移動させ、エコー高さがあるレベルを越える範囲の距離を欠陥寸法とする方法である。基準となるレベルとして、最大エコー高さより何dBか低いレベルを用いる場合と一定のエコー高さをを用いる場合とがあり、一般に前者の方法がよく用いられる。通常超音波探傷器のブラウン管は6dB下がったとき1/2の高さになるように表示されるため、例えば6dB down法を用いた場合、最大エコー高さの1/2の高さまでの範囲の長さが欠陥寸法となる。図2・8は6dB down法を用いた場合のもので、最大エコー高さ



表 2・4 超音波探傷による欠陥寸法の測定方法



(a) 水平方向の寸法



(b) 深さ方向の寸法

図 2・8 6 dB down法による欠陥寸法の推定方法

を  $H_m$  とすると欠陥寸法はエコーが  $H_m / 2$  以上の高さになっている範囲の長さとなる。(2)の方法はあらかじめ寸法のわかっている欠陥とそのエコー高さの関係を欠陥までの距離の関数として求めておき、未知の欠陥のエコー高さおよび欠陥までの距離から欠陥の寸法を推定する方法である。エコーの高さは欠陥の寸法だけではなく、欠陥の形状や傾き、表面粗さなどによって異なるため、この方法を用いて欠陥寸法を精度良く求めることは難しいといえる。欠陥の種類と寸法が推定できれば、それぞれの規格に基づき等級を調べ溶接の合否の判定を行うことになる。



## 2・2 エキスパートシステム

### 2・2・1 人工知能とエキスパートシステム<sup>(29)</sup>

エキスパートシステムとは、専門的に高度な問題の解決に関して、専門家と同等な能力をもつコンピュータシステムをいう。エキスパートシステムの分野では、そのようなコンピュータシステムを実現するために、マンマシン・システムの構築方法やその技術についての研究がなされており、人工知能(AI)の研究分野の一つの大きな柱となっている。

人工知能とは人間のように自分で考え行動する高度な知能を持った人工的な機械と見なすことができる。機械に思考させ、知的に行動させるというアイディアは20世紀に入って計算機が登場する以前からあったが、実際に計算機によって思考のシュミレーションを行い始めたのは1950年代に入ってからである。人工知能の研究は1956年に本格的に始まり、その後60年代前半まではパズルやボードゲームなどを樹木探索法により解く研究が盛んに行われた。しかし、探索技法を用いた場合、あるパズルに対してうまく解くことのできるプログラムは、他のパズルをうまく解くことができないという欠陥を持っていることがわかってきた。さらに、この探索技法は、人間が思考する方法ではなく、計算機に合った数理的方法で問題を解くものであったことから、現実問題の解決や人間の思考モデルに適用することがほとんど不可能であることもわかってきた<sup>(30)</sup>。その結果、人工知能の研究は一旦下火になった。

60年代後半になると、エキスパートシステム第一号ともいえるべきDENDRAL<sup>(4)</sup>が発表され、それから70年代にかけては知識工学の発展と共に多数のエキスパートシステムの研究が行われた。この知識工学の手法は人間の思考が膨大な経験的な知識を利用して行われ

ていることから考え出されたものである。エキスパートシステムは、この知識工学の考え方を取り入れたシステムであり、計算機の記憶装置に大量の知識を取り込み、これらを推論機構によって組合せ最終的に結論を導くというものである。エキスパートシステムの成功によって、人工知能の研究が再び盛んに行われるようになったが、その目的が単純で汎用性の高い推論機構をもった高度な知能の構築から人間の物真似をする機械を作ることにレベルダウンさせたとも言える。70年代以降に開発されたエキスパートシステムとして、HEARSAY-II<sup>(5)</sup>、MYCIN<sup>(6)</sup>、CASNET<sup>(7)</sup>、MACSYMA<sup>(8)</sup>、CADUCEUS<sup>(9)・(10)</sup>などがある。対象分野は医療診断が多く、上記のシステムの中ではMYCIN、CASNET、CADUCEUSなどがある。また、エキスパートシステムは知識の部分を他の知識に置き換えることによって異なる分野の問題を対象とするシステムとなる。このことから、エキスパートシステムの知識の部分だけが除かれたエキスパートシステム構築用ツールが開発されるようになっていった。

80年代に入ると実用化に向けての研究が盛んに行われるようになり、現在までに様々な分野におけるエキスパートシステムが開発され一部実用化されている<sup>(3)</sup>。

### 2・2・2 知識工学とエキスパートシステム<sup>(31)</sup>

初期の人工知能の研究者の汎用性の高い推論機構の構築という試みが不成功に終わったことから、それに代わって人間の思考方法から得られた知識工学の手法が開発された。人間は与えられた問題に対して常に一定のアルゴリズムを用いて解決しているわけではなく、自分の知識を利用してその中から最も適当と思われる方法を見つけ出しているのである。したがって、人間の思考の中で最も重要なものは思考過程ではなく蓄えられている知識であるといえる。そのために知識そのものの重要性に着目し、大量の知識を記憶させた知識ベースとその知



識を利用して推論を行う推論エンジンからなる知識ベースシステムが考え出された。このシステムは与えられたデータとシステムに蓄えられている知識とを照合することにより結論を導くものである。システムに蓄えられている知識が専門家から獲得したものであり、かつ専門家と同等の解決能力をもつ場合、そのシステムはエキスパートシステムと呼ばれる。図2・9はエキスパートシステムの構造を示したもので、使用者とシステムとのやり取りを行うユーザインタフェース、専門知識のデータベースである知識ベース、推論を行う推論エンジン、入力データや推論結果を記録するワーキングメモリ、推論結果が導かれた過程を説明する説明モジュールから成っている。

知識工学の分野では、得られた知識の表現方法や知識の獲得方法、知識を利用した推論方法などについての研究がなされている。知識の表現方法は対象とする問題の推論アルゴリズムに大きく関係するため、エキスパートシステムを構築する場合、どの知識表現モデルを用いるかを十分検討する必要がある。

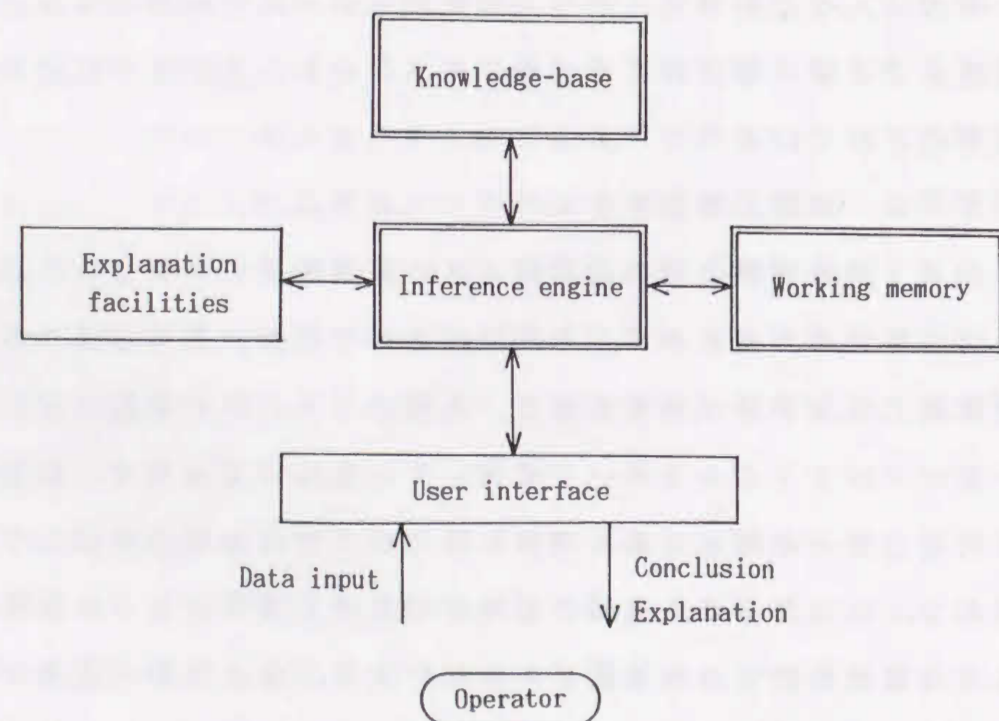


図2・9 エキスパートシステムの構造

### 2・2・3 知識表現モデル<sup>(22)</sup>

これまでに開発された知識表現モデルは、ルールモデル、ブラックボードモデル、因果ネットワークモデル、意味ネットワークモデルおよびフレームモデルに大別することができる。以下に、それらの知識表現モデルについて簡単に説明する。

ルールモデルは知識をルールの集合で表現したもので、代表的なものにIF-THEN形式のプロダクションシステムがある。例えば、「鳥は飛ぶ」という知識を考えてみよう。この知識をIF-THEN形式で表すと「if 鳥 - then 飛ぶ」となる。ここでデータとして「雀は鳥である」が与えられたとする。このデータと上記の知識とを照合することによって「雀は飛ぶ」という結論が得られることになる。このルールモデルは比較的簡単に知識を表現することができ、プロットタイプを構築する際には有用である。しかし、全てのルールに対して順番に照合していくため、ルール数が多くなると推論時間が長くなることや知識の全体構造が分かりにくいという欠点がある。さらに、上記のような三段論法的な推論しかできないため、推論に柔軟性がないという問題もある。

上記のルールモデルの欠点を解消するために、ルールおよびデータベースを構造化したブラックボードモデルがある。このモデルは、推論を階層化し、各階層毎にグループ化した知識を結び付けることにより、個々の推論に必要な知識のみを利用しようとするものである。それにより、不必要な推論を行わなくてよいことから推論時間が短縮されると共に、知識の全体構造が理解しやすくなる。

ルールモデルでは個々の知識はお互いに独立したルールとして表されるが、人間のもつ知識は独立ではなくお互いに依存し合っている。そのような相互依存性の知識に着目したモデルとして因果ネットワークモデルがある。このモデルはノード（節点）が状態や概念を表し、ブランチ（枝）が因果関係に対応し、ネットワーク全体として、ある



特定の意味を持つ知識の体系が表現される。しかし、このモデルは各状態間の因果関係が明確な問題に対しては有用であるが、そうでない場合には適用できないという欠点がある。また、ネットワークに特定の意味を持たせるようにしたデータ構造をもつ意味ネットワークモデルがある。因果ネットワークモデルは意味ネットワークモデルの一つといえる。

人間の知識構造により近い知識表現モデルとしてフレームモデルがある。人間は場面や環境によって用いる知識を変えている。例えば、仕事において用いる知識と遊びの時に用いる知識は全く別のものであるといえる。これは、人間は知識をある枠組によって整理しているといえる。そのように知識をある枠組（フレーム）によって表現する方法としてフレームモデルが考え出された。このモデルでは知識はグループ化され、それらの個々のグループはフレームとしてまとめられ、さらに個々のフレームはフレーム間の関係から抽象—具体関係もしくは全体—部分関係に階層化されている。このモデルは対象とする問題を表現することに対して非常に柔軟でありかつ強力であるが、十分に体系化し利用することは決して容易ではない。

上記の知識表現モデルのうち、どれがそのシステムに適しているかは、対象とする問題やシステムの構築の目的によって異なる。また、ブラックボードモデルにおいて知識のグループ化にフレームモデルを用いるなど、いくつかのモデルを統合したシステム作りも考えられる。

#### 2・2・4 開発されたエキスパートシステムの特徴

ここでは70年代以降に開発されたエキスパートシステムの中からMYCIN, CASNET, CADUCEUS, HEARSAY-II, MACSYMAについて簡単に説明する。

MYCINは伝染性血液疾患の問題を扱っており、知識は400個ほどのルールより成っている。MYCINは単純なIF-THEN形式の個々に独立した規則を利用した最初のシステムであり、この方法

はその後の知識表現の研究に大きな影響を与えた。また、個々のルールの不確実さを確信度と呼ばれる-1（確実な誤り）から1（確実な真実）までの間の数値を用いて表現し、不確実性を取り扱える推論方法を開発している。この方法はMYCINの方法として、その後の多くのシステムに用いられると共に、不確実性を取り扱ったシステムの研究に大きな影響を与えた。このMYCINからエキスパートシステム開発用ツールとしてEMYCINが開発され、PUFFといったエキスパートシステムが開発されている。

CASNETは緑内障の診断と治療を対象にしたシステムで、因果ネットワークモデルを用いることにより疾患の因果関係をモデル化している。不確実さを扱うためにネットワークの中のリンクには重みを与えられている。その推論は病気の因果連鎖モデルの下でのデータ駆動による解釈を利用することにより行っている。また、医学の参考情報を数多く含んでおり、推論の説明機能が充実している。

CADUCEUSは内科の分野での疾患の診断を試みたシステムで、この分野の知識の約85%を表現している意味ネットワークより成っている。手順は仮説に基づいてデータを集めるための質問をして、いくつかのデータから病気に対する可能な診断仮説を導くというものである。このシステムの知識は、約500の病名、約350の病状、約10万の徴候との連鎖からなっており、エキスパートシステムの知識ベースとしては最大であるとされている。

HEARSAY-IIは音声理解システムとして構築されたものである。このシステムでは音声を理解する作業を階層化し、知識はその階層毎にグループ化し、独立した知識モジュールとして構成されている。個々の独立した知識は黒板と呼ばれる大域的なメモリに結び付けられており、黒板を通じて推論が階層的に行われていくという新しい推論方法（ブラックボードモデル）を用いた。これによって、不必要な再計算を行わなくてもすむようになり、かつ推論の流れが大局的に理解



しやすくなったという利点を得られた。この新しい推論方法はその後  
の知識工学の研究やエキスパートシステムの開発に大きな影響を与え  
た。H E A R S A Y - II からエキスパートシステム構築用ツールとし  
てH E A R S A Y - III が開発されている。

M A C S Y M A は代数計算や積分などの数学的な問題を解くために  
開発されたシステムである。システムの専門知識は数学の研究者の経  
験的知識の集まりである。このシステムには独立した知識ベースはな  
く、知識はプログラムの一部として表現されている。また、優れたユ  
ーザインタフェースを持っている。しかし、基本原理ではなく専門家  
の経験的知識のみを利用していることから幅広い問題を扱うことがで  
きないことや推論の説明機能がない、不確実さを取り扱うことがで  
きないといった問題がある。

## 2・3 非破壊試験を対象としたエキスパートシステム

溶接の良否を判定する非破壊試験を対象としたエキスパートシステ  
ムの開発も盛んに行われている<sup>(11)-(17), (33)-(38)</sup>。ここでは、こ  
れまでの開発事例のいくつかを紹介する。

### 2・3・1 超音波探傷試験による溶接欠陥の検出と

評価のためのエキスパートシステム<sup>(11)-(13)</sup>

このエキスパートシステムは超音波探傷試験の試験方法の選定を行  
うシステムと試験結果の評価を行うシステムの2つから構成されてい  
る。前者は、最適な超音波探傷試験方法を決定するためのシステムで  
あり、後者は溶接欠陥の種類と大きさおよび欠陥等級を推論するもの  
である。

前者のシステムはマイクロコンピュータ上のエキスパートシステム  
構築支援ツールE X S Y Sを使用したもので、I F - T H E N 形式ブ

ロダクションルールによって知識を表現している。このシステムの知  
識はW E S 2 0 2 1 を基にしているが、この知識が必ずしも現場にお  
いて最適とならない場合がある。その場合には、現場で培われた検査  
技術者の経験を聞き取り調査によって獲得し、その情報を知識データ  
ベースに組み込んでいる。また推論方法として後向き推論を採用する  
ことにより、推論時間の短縮が図られている。このシステムによって  
選択された試験方法と他の試験方法による結果を比較することによ  
って、この試験方法を用いた場合が最も精度良く欠陥を評価できるこ  
とが示されている。

後者の溶接欠陥評価のためのエキスパートシステムでは、市販のエ  
キスパートシステム構築用ツールの代わりにL i s p 言語を用い独自  
のプログラムを開発している。市販のツールでは特定の知識表現モデ  
ルが採用されていることから、知識を柔軟に表現することができない  
ため、超音波探傷試験の結果を評価するために必要な知識を表現す  
るには十分でなかったものと考えられる。このシステムではプロダクシ  
ョンルールを構造化した黑板モデルを採用し、プロダクションルール  
はフレームによってグループ化されている。このシステムが推論した  
結果と実際の欠陥の状況を比較し良い一致を示している。

### 2・3・2 放射線透過写真の自動合否判定システム<sup>(14), (15)</sup>

このシステムは放射線透過試験による欠陥の判定を自動的に行うも  
のである。放射線透過試験とは被検査物の放射線透過写真を撮影し、  
その写真の像から欠陥の有無および欠陥状況を推定し溶接の良否を判  
定するものである。このシステムは、欠陥の判定に必要な情報を写真  
から抽出する画像処理部と画像処理により得られた情報から欠陥の判  
定を行うエキスパートシステム部とから成っている。画像処理部では  
I T V カメラによって入力された写真の像から欠陥候補像の抽出を行  
い、欠陥候補像の形状や濃度分布の特性が計算される。エキスパート  
システム部には、画像処理部で得られた情報から欠陥の評価を行うた



めの知識がプロダクションルール形式によって表現されている。また、知識に含まれているあいまいさについては確信度を用いて表現できるようになっている。

### 2・3・3 超音波探傷試験結果評価支援エキスパートシステム<sup>(16)</sup>

原子力発電所では原子炉を初めとしてそれに接続される配管等の溶接部は定期的に非破壊検査が実施されており、この検査は「供用期間中検査」と呼ばれている。このシステムは原子炉に付属する配管の溶接部およびその近傍の母材の超音波探傷試験の結果の評価を支援するもので、ラップトップパーソナルコンピュータJ3100上のエキスパートシステム開発ツール「知的日本語プログラム言語T O O L」を用いて構築されている。このシステムでは、評価を難しくする妨害エコーの判定も行えるようになっている。また、供用期間中検査に用いるため応力腐食割れについても判定できるようになっている。さらに、波形をパターン化して表示し対話機能の充実に努めていると共に結果に至る経過やその理由を表示する機能を持ち初心者の教育にも使用できるように配慮されている。

### 2・4 非破壊試験を対象としたエキスパートシステムの問題点

これまでに開発された非破壊試験を対象としたエキスパートシステムの問題点について以下に簡単に説明する。

エキスパートシステムの能力は獲得された知識の質と量に依存するため、知識の獲得は非常に重要な問題となる。知識の獲得とは対象とする問題領域から専門知識を獲得し、知識ベースを構築することである。専門知識を獲得する場合、通常専門家からの聞き取りによって行うことが多く、知識工学者と専門家との共同作業によって行われる。その際、知識工学者がその専門分野に全く無知である場合、専門家からうまく知識を聞き出すことは不可能である。さらに、専門家自身も

意識していない経験的知識を持つ場合も考えられ、そのような知識が知識工学者との作業中に引き出されるように工夫する必要がある。しかし、満足のいく知識獲得ができることは難しく、一般にエキスパートシステムではこの知識獲得が最も重要な問題となっている。

また、専門家の持つ知識はあいまいであることが多い。したがって、エキスパートシステムを構築する際、そのようなあいまいな知識をどのように扱うかが問題となる。しかし、これまでの非破壊試験を対象としたエキスパートシステムでは、これらのあいまいさを十分に扱えるものはほとんど無いのが現状といえる。



### 3・1 ファジィエキスパートシステムの現状

#### 3・1・1 ファジィ理論

ファジィ理論の概念はカリフォルニア大学の Z a d e h 教授によって1965年に提唱されたものである<sup>(39)</sup>。同教授は実社会に存在するもののほとんどすべてが、通常の集合論で述べられるようにはっきりと性質を区別することができないと考え、あいまいなものを表現する方法として特性曲線の一般化といえるメンバシップ関数を導入したファジィ集合論を提案した。その後、同教授はクリスプなアルゴリズムをファジィ化したファジィアルゴリズムの有用性についても提唱している<sup>(40)・(41)</sup>。特に、人間が中心となるシステムや非常に複雑なため数学モデルを記述することが困難なシステムを扱う場合にファジィアルゴリズムが適していることを提唱した。このファジィ理論は、1970年代に入ると工学の研究者や技術者の関心を集めるようになっていった。このファジィ理論が最も早く適用されたのは制御の分野であった。世界で最初のファジィ制御は、M a m d a n i によってスチームエンジンの模型の制御として実証的に適用され<sup>(42)・(43)</sup>、その後セメントキルンを対象としたファジィ制御が開発されて実際の機械の自動運転に応用された<sup>(44)</sup>。このような種々の分野でのファジィ制御の成功により、ファジィ理論はファジィ制御の分野において広く適用されるようになっていった<sup>(45)</sup>。

最初に、ファジィ集合について簡単に説明する。いま「風呂の湯が熱い」の「熱い」を考えてみる。この場合、例えば42℃というようなある数値を用いて、それ以上の時熱く、それ以下では熱くないという具合に明確に分けることは不可能である。一人の人間でも常に同じ

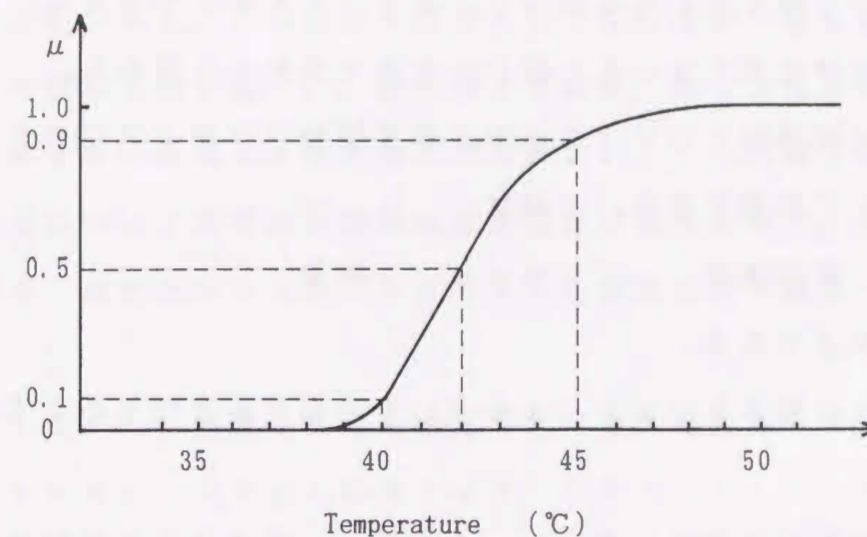


図3・1 熱く感じる湯温のメンバシップ関数

温度で熱いと感じるとは限らないし、多くの人を考えた場合、人によって熱く感じる温度は異なってくる。したがって、熱い風呂の湯の温度は通常の集合論で用いられる1もしくは0の不連続な特性曲線では表現することができない。このような場合、ファジィ集合では1もしくは0のみの特性曲線ではなく、1から0の連続な特性曲線（メンバシップ関数）を用いて図3・1のように表すことができる。この関数は風呂の湯が熱いとみなせる度合を表したもので、40℃の時0.1、42℃の時0.5、45℃の時0.9となる。この0.1や0.5という数値がその度合を表している。このことから、ファジィ集合とは通常の集合の拡張であるといえる。

次に、ファジィ制御におけるファジィ推論について簡単に説明する。まず、人間が自動車を定速運転させる場合を考える。このとき、他の車の影響は受けないものとする。この時の操作としては以下のようなものが考えられる。

- ・速度が非常に早くなったと感じたときアクセルを弱める。
- ・速度が少し早くなったと感じたときアクセルをやや弱める。



- ・速度が一定と感じるときアクセルはそのままとする。
- ・速度が少し遅くなったと感じたときアクセルをやや強める。
- ・速度が非常に遅くなったと感じたときアクセルを強める。

以上の操作は制御則として I F - T H E N 型のファジィプロダクションルールとして表現できる。これは

i f      前提条件      t h e n      結果

で表される形式である。

上記の一番目の操作をこのルールを用いて記述すると以下のようになる。

i f    速度が非常にはやい      t h e n    アクセルを弱める

このようなルールは一般的に以下のように表現できる。

R 1 :    i f      x = A 1      t h e n      y = B 1

R 2 :    i f      x = A 2      t h e n      y = B 2

・

・

・

R n :    i f      x = A n      t h e n      y = B n

このとき  $A_i$ ,  $B_i$  ( $i = 1, 2, \dots, n$ ) はファジィ集合である。

以下にファジィ推論について M a m d a n i が用いた推論方法<sup>(42)</sup>を用いて説明する。観測結果として  $x = A_0$  が得られたとすると、それぞれのルールの前件部の条件に観測結果がどの程度適合するかを示す一致度を求める必要がある。その一致度をファジィ集合  $A_i$  の  $A_0$  におけるメンバシップ値とすると、

$$\omega_i (\text{一致度}) = A_i (A_0)$$

となる。次に、 $i$  番目のルールの推論結果を

$$y_i = \min(\omega_i, B_i)$$

とする。全てのルールによる推論結果  $y_0$  は

$$y_0 = \max(y_i)$$

とする。このようにして得られた  $y_0$  から制御するための代表値を求める必要がある。この操作を非ファジィ化と呼び、M a m d a n i は  $y_0$  の重心を用いた<sup>(42)</sup>。以上のことから、M a m d a n i の推論方法は M i n - M a x - 重心法と呼ばれている。この M i n - M a x - 重心法について前述の自動車の定速運転の操作を例に、図を用いて説明する。速度を 60 km/h とすると制御則は以下になる。

・ i f      速度が 60 km/h を大きく越える・・・P B i

t h e n    アクセル開度を小さくする・・・N B t

・ i f      速度が 60 km/h を少し越える・・・P S i

t h e n    アクセル開度をやや小さくする・・・N S t

・ i f      速度が 60 km/h ..... Z O i

t h e n    アクセル開度はそのままとする・・・Z O t

・ i f      速度が 60 km/h より少し低い・・・N S i

t h e n    アクセル開度をやや大きくする・・・P S t

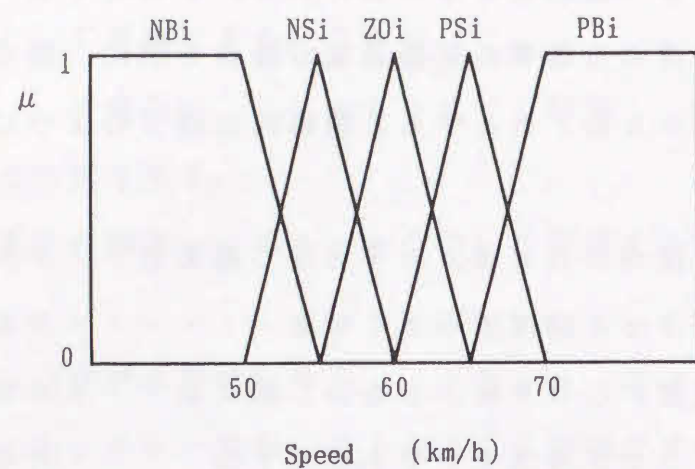
・ i f      速度が 60 km/h より非常に低い・・・N B i

t h e n    アクセル開度を大きくする・・・P B t

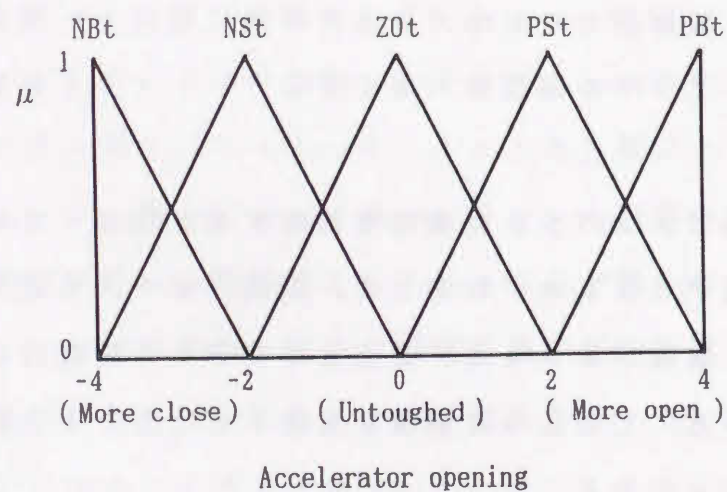
この制御則を基にメンバシップ関数を表現すると図 3・2 のようになる。ここで速度が 58 km/h のとき、制御プロセスを図 3・3 に示す。これより、速度が 58 km/h のときアクセル開度は 0.84 になることがわかる。このように制御を反復することにより自動車は定速で走行することになる。

ファジィ制御の推論方法として M a m d a n i の M i n - M a x - 重心法の他に代数積 - 加算 - 重心法などがある。また、非ファジィ化の方法としては、重心法の他に中央値法や高さ法などが提案されている<sup>(3)</sup>。





(a) 前件部メンバシップ関数



(b) 後件部メンバシップ関数

図3・2 定速運転用メンバシップ関数

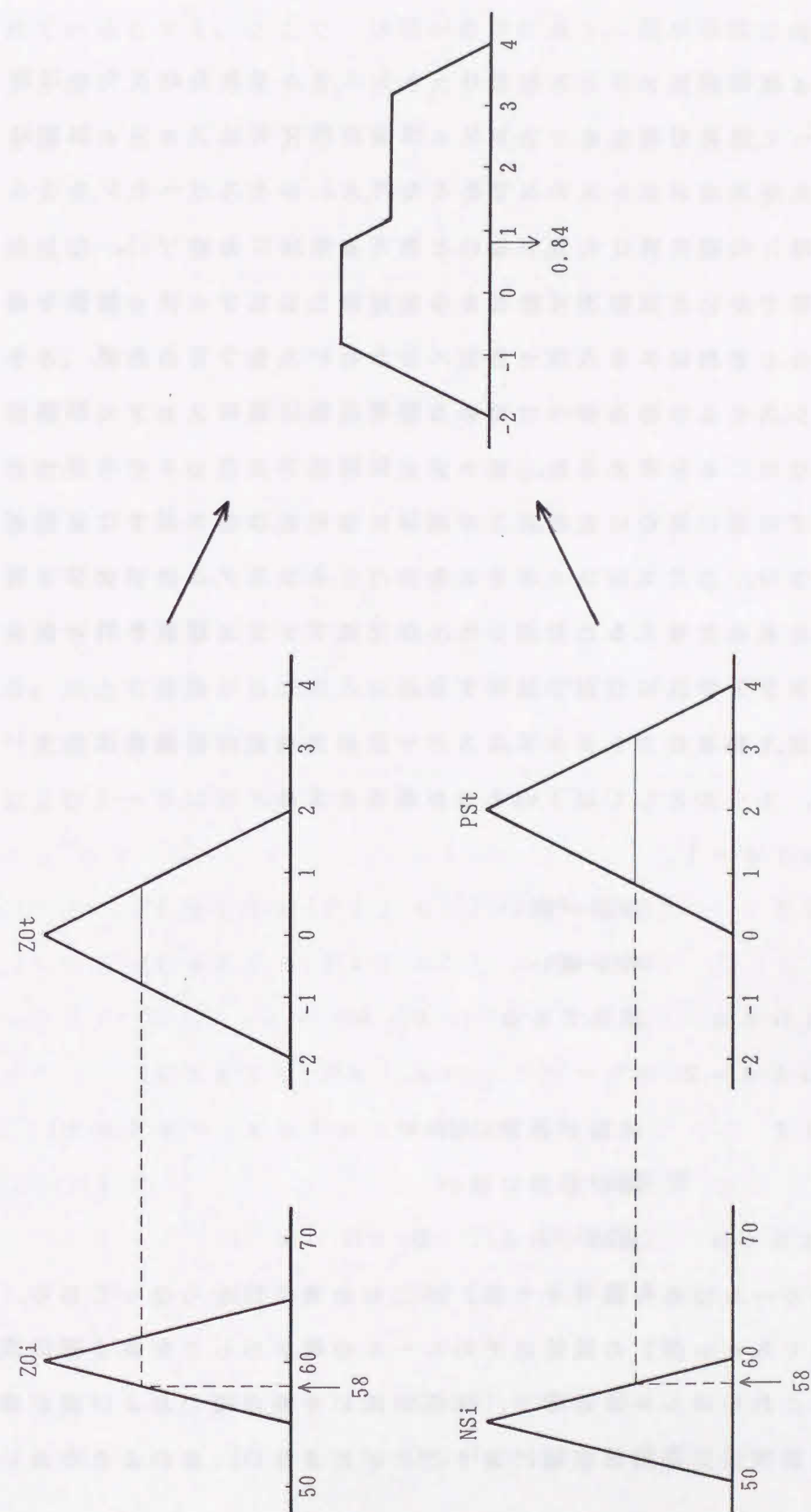


図3・3 ファジィ推論



### 3・1・2 ファジィエキスパートシステム

ファジィ制御がファジィエキスパートシステムであるかどうかは研究者によって意見が異なる。ファジィ理論の研究者はファジィ制御はファジィエキスパートシステムであると考え、エキスパートシステム（知識工学）の研究者はそうでないと考える傾向にある<sup>(45)</sup>。たしかに、専門家でないと制御不可能なものを対象としてファジィ制御を適用する場合、それはエキスパートシステムといえそうであるが、エキスパートシステムの特長の一つである説明機能は通常ファジィ制御には存在しないことを考えると、ファジィ制御はファジィエキスパートシステムではないともいえる。この議論についてはさまざまな意見があると思うが、ここではファジィエキスパートシステムはファジィ制御を除いたものと考えことにした。以下にファジィ理論を用いたエキスパートシステムについて説明する。

ここでは、簡単なファジィエキスパートシステムの推論方法について考える。ルールとして以下の2つがあるとする。

```
rule-1
  if      体温が高い
          喉が痛い
  then    風邪である      0.6
rule-2
  if      体温が非常に高い
          頭が非常に痛い
  then    流感である      0.7
```

これらのルールは条件部（if部）が二つの条件節からなっており、結論部（then部）の数値はそのルールの確からしさを表す確信度である。これらのルールの中で、体温が高いや喉が痛いおよび頭が痛いという表現は定量的に明確に表すことができない。このようなあい

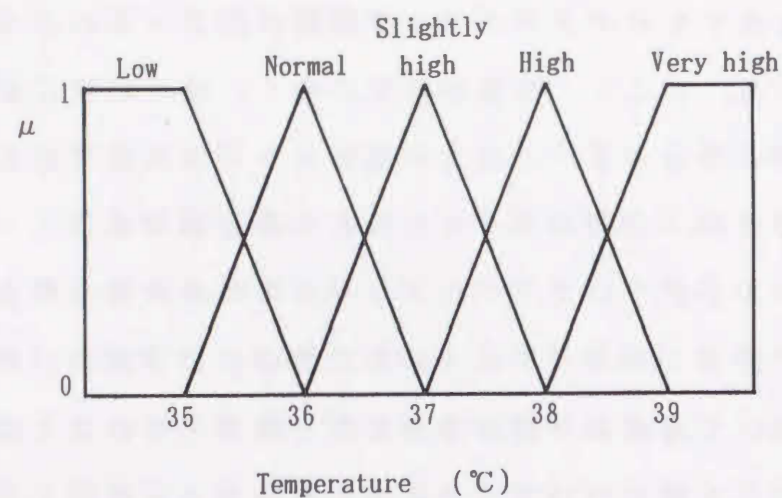
まいな表現を表すためのメンバシップ関数が図3・4のように与えられているとする。ここで、体温が非常に高く、頭が非常に痛く、喉が少し痛い人がいたとする。この人が風邪もしくは流感であるかどうかを調べるためには、上記のルールと照合する必要がある。

まず、ルール1について、データとルールの条件部の照合を行う。体温について図3・4（a）の「非常に高い」と「高い」のメンバシップ関数を用いて両者の一致度を求めると図3・5のように0.5となる。同様にして喉についても一致度を求める（一致度0.5）。頭痛についてはルール1の条件部にないため考えなくてよい。条件部の確信度としては各条件節の一致度の最小値となる（0.5）。最後に結論として条件部の確信度とルールの確信度との最小値より「風邪である可能性は0.5である」が得られる。ルール2についても同様な推論を行うことにより「流感である可能性は0.7である」が得られる。以上の推論からこの人は流感である可能性が高いことがわかる。

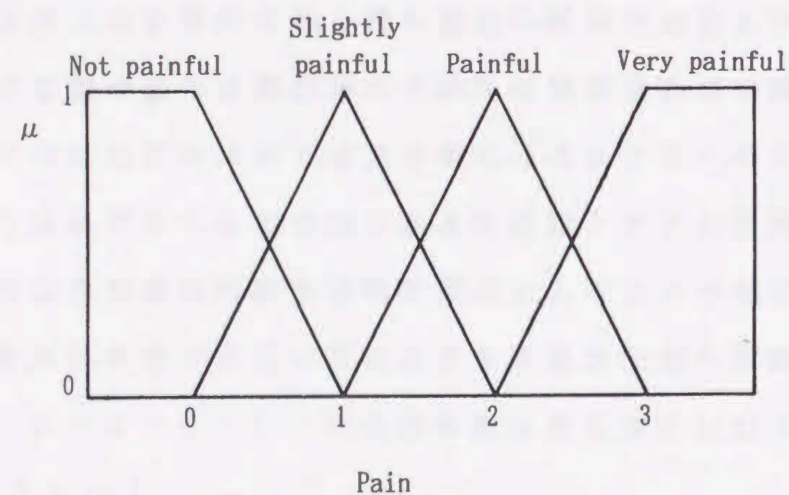
上記の推論を一般的に表現すると以下のようなになる。まず、データおよびルールは以下のように表される。

```
(( x1 is P1 ) u1 ) : データ1
(( x2 is P2 ) u2 ) : データ2
. . . . .
(( xm is Pm ) um ) : データm
(rule i rule-name
  (if                                     : 条件部
    (( x1 is Q1 )) : 条件節1
    (( x2 is Q2 )) : 条件節2
    . . . . .
    (( xn is Qn )) : 条件節n )
  (then (( yi is Ri ) vi)) : 結論部 )
```

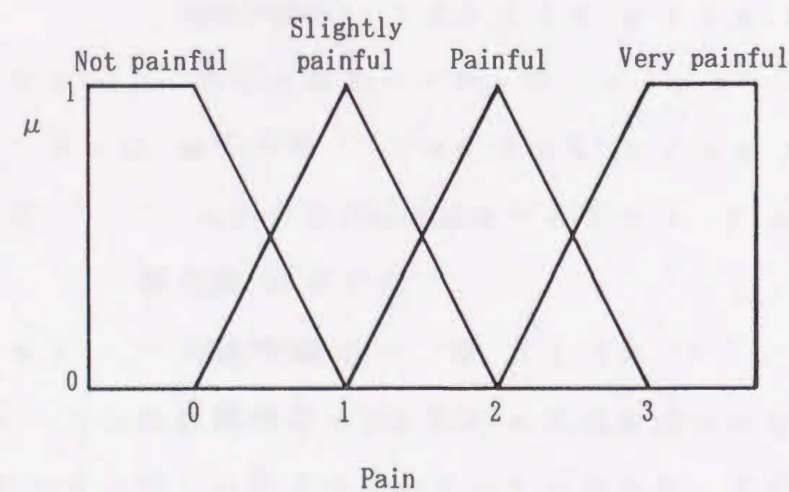




(a) 体 温



(b) 喉の痛み



(c) 頭 痛

図 3・4 メンバシップ関数

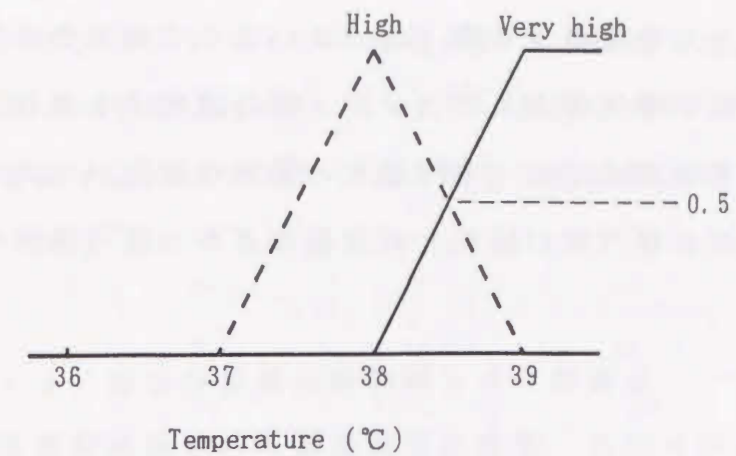


図 3・5 一 致 度

ここで、 $x_i$ は検査項目、 $y_i$ は仮説レベル、 $P_i$ 、 $Q_i$ および $R_i$ は検査項目 $x_i$ および仮説レベル $y_i$ に対するファジィ集合（メンバシップ関数）で定義された評価、 $u_i$ および $v_i$ はそれぞれデータおよびルール自身の確信度である。

推論手順は以下の通りとなる。まず、ルールベースから最初のルールを取り出す。その後ルールの条件部の各条件節と各データの一致度 $m_{xi}$ を求める。 $m_{xi}$ は、 $P_i$ および $Q_i$ のメンバシップ関数の交わりの高さとして求められる。次に、各条件節の確信度 $m_i$ を $m_{xi}$ と $u_i$ の最小値（論理積）として求める。さらに、条件部全体の確信度 $m$ は各条件節の確信度 $m_i$ の最小値（論理積）となる。条件部全体の確信度が求めれば、それとルール自身の確信度の最小値（論理積）が結論部の確信度となり、それが結論となる。最初のルールについて推論が終わると、ルールベースから次のルールを取り出し、そのルールについても推論を行う。この操作をすべてのルールに対して行うことにより各結論部に対する確信度が求められる。確信度の最も大きい結論が可能性の最も高い結論となる。

以上のことから、ファジィ理論を用いることにより知識や測定データに含まれるあいまいさや不確実さをメンバシップ関数や確信度によ



って表現し取り扱うことができることがわかる。しかし、ファジィエキスパートシステムを構築する際、あいまいさや不確実さを表現するメンバシップ関数や確信度およびファジィ結合演算子を適切に決定する必要がある。そのためには、専門家との緊密な話合いやエキスパートシステムの結果と専門家の結果との比較などから試行錯誤を重ねる必要がある。

### 3・2 プロットタイプの構築

#### 3・2・1 はじめに

各種溶接構造物の溶接欠陥の検査として非破壊検査が行われるが、その中で超音波探傷試験は作業性の良さや装置の性能の向上などの理由により、建築物などに広く用いられるようになってきている。この超音波探傷試験の結果を評価することにより、溶接欠陥の種類、欠陥寸法および欠陥等級などを推定し、欠陥の可否の判定を行うことができる。しかしながら、建築物などの超音波探傷試験では小型の探傷器が用いられるため、情報量の少なさからこれらの推定にはかなりの熟練を必要とされる。また、検査技術者によって主観や推定方法が異なるため、推定結果の差異が問題となる場合も少なくない。さらに、近年は労働力不足から超音波探傷試験の検査技術者を養成することが難しくなっていることが、この問題に拍車をかけている。

これらのことより、精度の高い超音波探傷試験評価システムを構築することは有益なことであり、すでにエキスパートシステムの開発が報告され良好な成果が挙げられている<sup>(11)・(47)</sup>。しかしながら、超音波探傷試験の知識や測定データに存在するあいまいさについてはほとんど考慮されていない。そこで本節では、超音波探傷試験の検査技術者が持つあいまいな専門知識やあいまいなデータをファジィ集合を

用いて表現し、そのような知識やデータについての推論が行えるようなシステムのプロットタイプの構築を行う。最近では、工学分野だけに限らず多くの分野において、ファジィ理論を用いてあいまいさを扱おうとするエキスパートシステムの開発の成功例が数多く報告されている<sup>(48)・(54)</sup>。

#### 3・2・2 超音波探傷試験評価とその問題点

超音波探傷試験結果から溶接欠陥の種類、欠陥寸法および欠陥等級の推定を行うが、その中でも特に欠陥種類を正確に推定することは欠陥寸法や可否判定などの際に重要となる。欠陥種類の判定には種々の方法が提案されているが、欠陥種類を精度良く判別し得るものはいまだ完全には確立されていないのが現状といえる。したがって、欠陥種類の推定に用いられる方法は検査技術者によって異なり、また推定結果は検査技術者の主観によって大きく左右されることは否めない。図3・6は、欠陥種類の推定方法の一例を示したものである<sup>(55)</sup>。この方法では、まず欠陥発生部位を推定し、その後フローチャートにしたがって判定を行うことにより最終的に欠陥種類が導かれる。判定部の、〈Difference of echo height〉は両側探傷した時のエコー高さの差が10dB以下かどうかの判定、〈Beam path distance〉は色々なビーム路程でエコーが立ち上がるかどうかの判定、〈Shape of echo〉はエコーの波形がギザギザかどうかの判定、〈Distance amplitude correction curve〉は距離振幅特性曲線がA2-φ4より緩やかかどうかの判定、〈FP〉は材質や溶接方法、開先形状等による欠陥の発生特性を判断することをそれぞれ意味している。また、図中中央の判定部のTは、表3・1に示す項目について探傷結果の該当する部分に与えられている点数の合計として計算される。ところで、欠陥発生部位を推定する際に、溶接の施工精度や溶接の溶け込み量等を正確に把握することが困難なことおよび測定誤差の問題等から、欠陥がルート部、溶接金属部、



表 3・1 探傷結果の評価

Estimation item	Class	Points	Class	Points	Class	Points
Echo height	< H line	1	H line~ H line	2	> H line	3
Lateral scan	< 10mm	1	10~30mm	2	> 30mm	3
Orbital scan	< 8dB	1	8~15dB	2	> 15dB	3
Depth scan	< 10mm	1	10~20mm	2	> 20mm	3

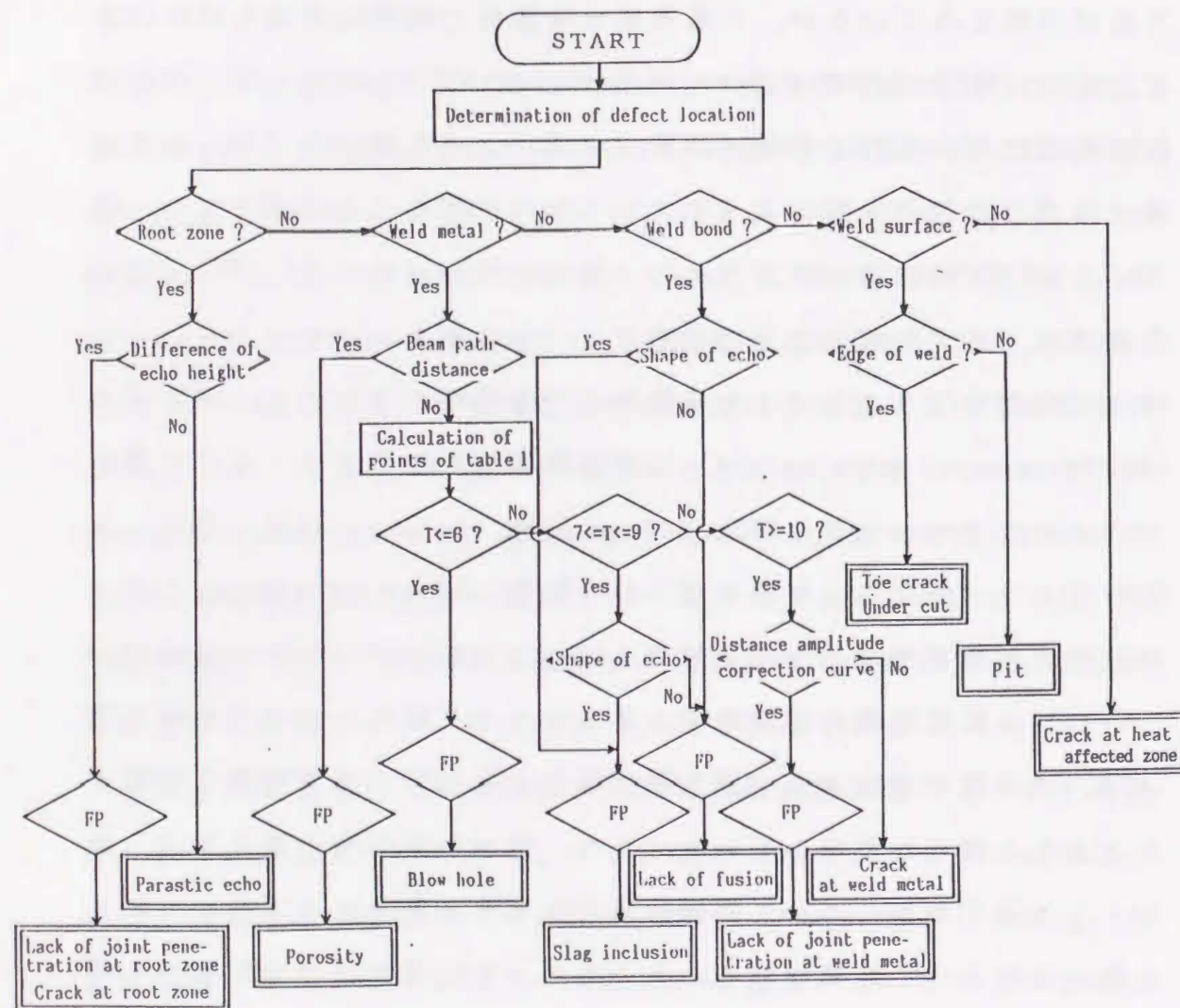


図 3・6 斜角一探触子法による溶接欠陥種類判別フローチャート

開先部、HAZ部（熱影響部）あるいは母材部のどこに位置するかを（特にそれらの境界付近に位置する場合）限定することは非常に困難となる（図 3・7）。また、フローチャート中の判定部においても、〈Yes〉もしくは〈No〉のどちらかに決められない場合も少なくない。そこで、このようなあいまいさをメンバシップ関数および確信度を用いて表してファジィ理論を適用することにする。

### 3・2・3 メンバシップ関数および確信度

本システムで用いたメンバシップ関数および確信度<sup>(31)</sup>は現場で培われた非破壊検査技術者の経験を参考に求めたものである。それらについて以下で説明する。

(1) 欠陥発生部位推定用のメンバシップ関数 溶接の溶け込み量は、溶接の電流や電圧値、溶接棒と溶接材との角度および距離等によって異なる。しかしながら、これらのことを全て考慮することは非常に困難なので、ここでは現場で使用されている経験的な値を用いメンバシップ関数の形状を与えた。図 3・8 に V 形開先の場合の欠陥発生部位推定用のメンバシップ関数の形状を示す。溶け込み量は 1 ~ 3 mm, HAZ 部は開先面から 2 ~ 5 mm という値を基に作成した。

(2) データ判定用のメンバシップ関数 図 3・6 のフローチャート中の欠陥発生部位以外のデータの判定に用いるメンバシップ関数について説明する。

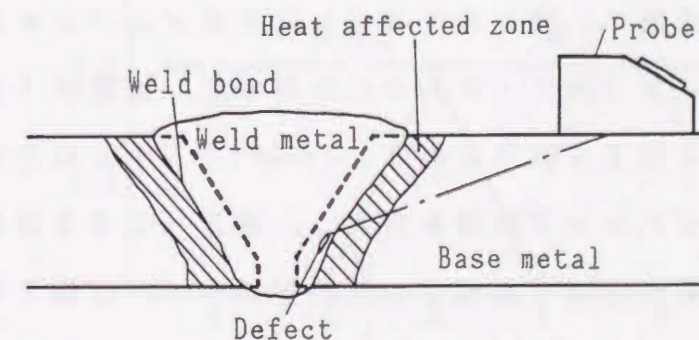


図 3・7 溶接部の断面



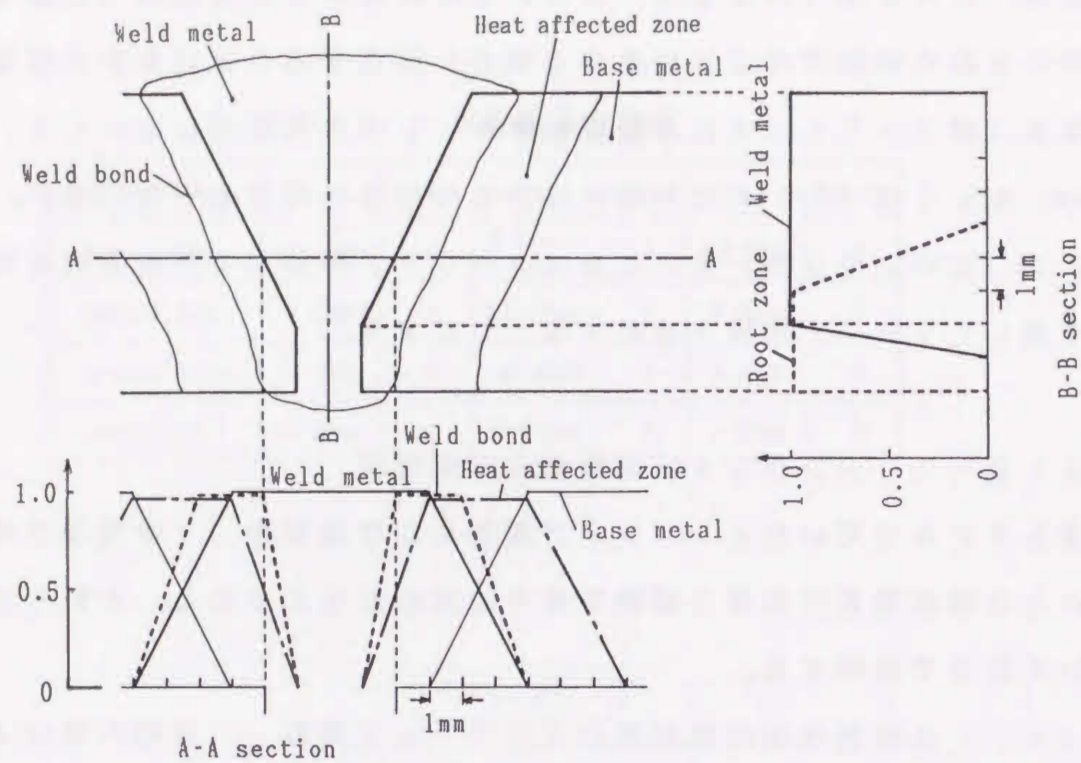
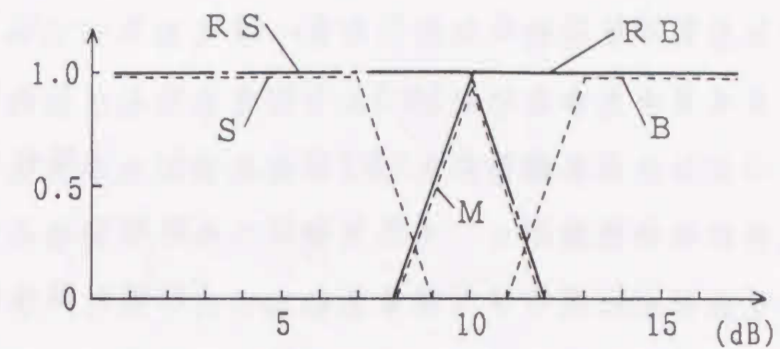


図3・8 欠陥発生部位推定用メンバシップ関数 (V形開先)

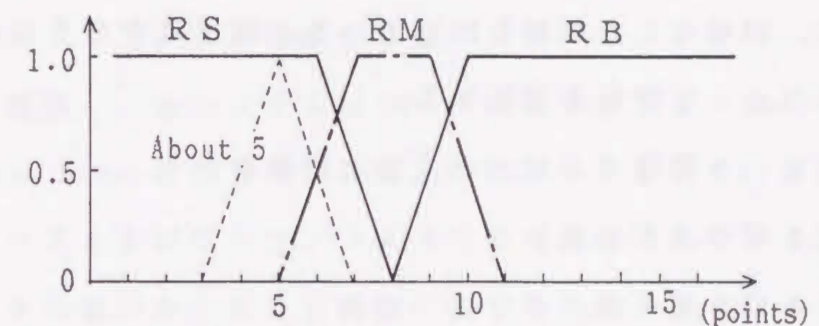
(a) 両側探傷した時のエコー高さの差は10 dB以下か  
図3・9(a)にそのメンバシップ関数を示す。フローチャート(図3・6)にしたがって「10 dB以下である」(RS), 「10 dBより大きい」(RB)のメンバシップ関数と、入力データに対応する3つのメンバシップ関数「10 dBより明らかに小さい」(S), 「10 dB前後である」(M), 「10 dBより明らかに大きい」(B)を定義する。

(b) 表1の点数 図3・9(b)にそのメンバシップ関数を示す。フローチャート(図3・6)にしたがって、点数が「6以下である」(RS), 「7以上9以下である」(RM), 「10以上である」(RB)の3つのメンバシップ関数を定義し、表3・1より計算された点数は、その点数を中心とした幅2の三角形のファジィ数として与えた(例として点数が5の時のメンバシップ関数を図に示す)。

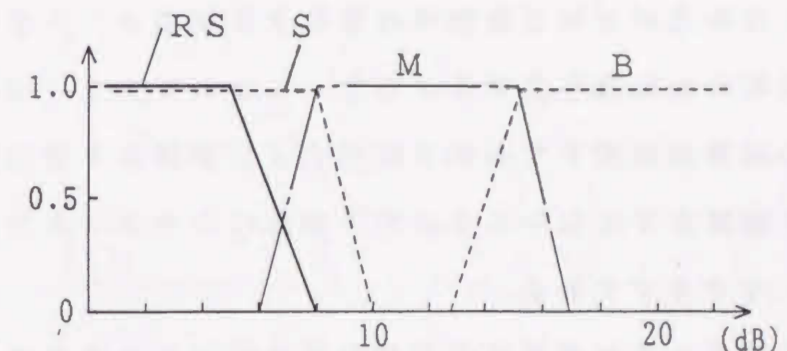
(c) 振子走査した時のエコーの低下は小さいか これは欠



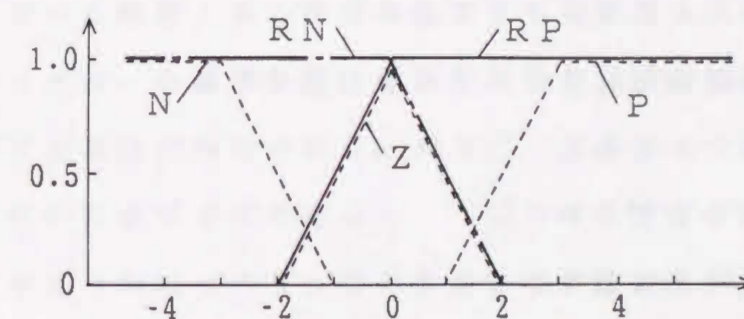
(a) 両側探傷した時のエコー高さの差



(b) 表1の点数



(c) 振子走査した時のエコー高さの差



(d) その他の判定部

図3・9 メンバシップ関数



陥がブローホールの場合に、振子走査した時のエコーの低下が小さいことを利用して判定に用いるものである。図3・9(c)に「小さい」(RS), 「8 dBより小さい」(S), 「8 dB以上15 dB以下である」(M), 「15 dBを越える」(B)のメンバシップ関数を示す。

(d) その他の判定部 「色々なビーム路程でエコーが立ち上がるか」, 「エコーの波形はギザギザか」, 「距離振幅特性曲線はA2-φ4より緩やかか」の判定については同じ形状のメンバシップ関数(図3・9(d))を用いる。横軸は数値化できない感覚的な変数であるので、規格化した数値を用いている。以下にそれぞれの判定に対するメンバシップ関数を説明する。

色々なビーム路程でエコーが立ち上がるか

RN: 立ち上がらない

RP: 立ち上がる

N : 明らかに立ち上がらない

Z : 立ち上がるかどうかははっきりしない

P : 明らかに立ち上がる

エコーの波形はギザギザか

RN: ギザギザでない

RP: ギザギザである

N : 明らかにギザギザでない

Z : ギザギザかどうかははっきりしない

P : 明らかにギザギザである

距離振幅特性曲線はA2-φ4より緩やかか

RN: 緩やかである

RP: 緩やかでない

N : 明らかに緩やかである

Z : 緩やかかどうかははっきりしない

P : ほぼ同じか急である

(3) 確信度 知識自身の確からしさを表すために0~1の間の確信度と呼ばれる値がよく用いられる。本システムにおいても確信度を用い、知識自身に含まれるあいまいさを考慮できるようにした。例えば、超音波探傷試験評価による欠陥種類の判定においては、割れ(溶接金属内部割れ, ルート割れ, ビード下割れ等)の発生頻度が小さいことから、これらの欠陥種類を導くルールの確信度を小さく(例えば0.4)した。

### 3・2・4 システムの構成と手順

(1) 概要 本システムはワークステーション(Apollo 3500)上に、Common Lisp 言語を用いて独自に開発したもので、推論エンジン、知識ベースおよびワーキングメモリから構成されている。通常、知識やデータの量が多くなると推論に要する時間が長くなるので、この問題を解消するために本システムでは知識やデータをグループ化する黑板モデル<sup>(31)</sup>を採用した。また、知識はプロダクションルール形式とし、フレームモデルによってグループ化した。ルール数は269個で、推論に要する時間は入力時間を含めて1例あたり約3分である。図3・10に本システムの黑板モデルを示す。以下に推論手順を述べる。

まず、溶接条件および超音波探傷試験結果のデータを対話式に入力する。このとき、入力項目は、「板厚」, 「開先角」, 「探触子の屈折角」, 「ビーム路程」等の数値を入力する項目と、「開先形状」, 「材質」, 「エコーの波形」, 「距離振幅特性曲線の傾き」等の選択肢から選択する項目に分けられる。例えば、「エコーの波形」については、「明らかにギザギザである」, 「明らかにギザギザでない」, 「ギザギザかどうかははっきりしない」の3つの中から選択することになる。

入力が終わると、溶接条件データより知識 Weld\_rules を用いて欠



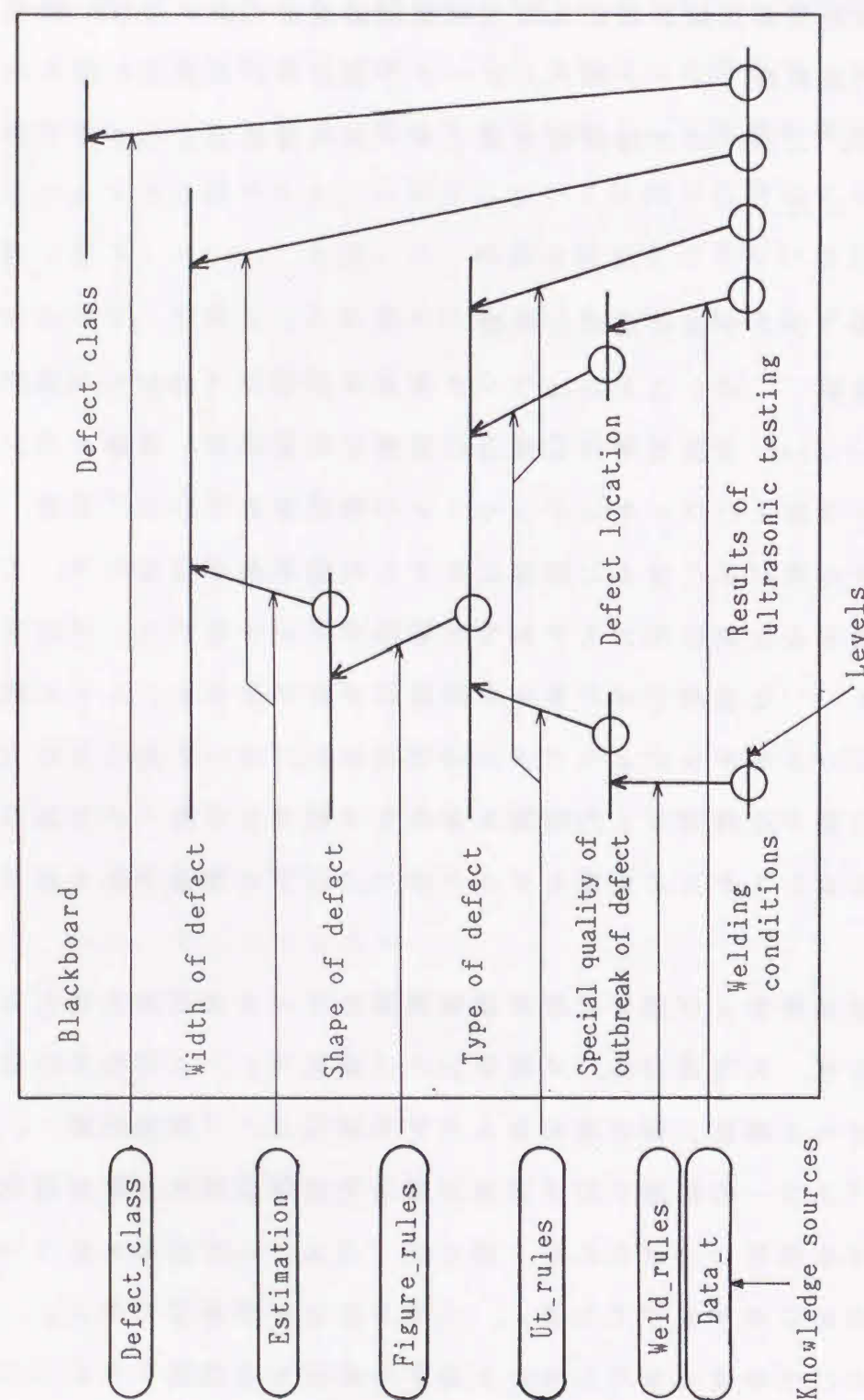


図3・10 黒板モデル

陥発生特性を，超音波探傷試験結果のデータより知識 Data\_t を用いて欠陥発生部位を推論する。欠陥発生特性は材質，溶接方法，開先形状等の違いによる各々の欠陥の発生頻度を表している。欠陥発生部位は，欠陥がルート部，溶接金属部，開先部，HAZ部および母材部の各々の部位に位置する可能性をメンバシップ関数（図3・8）よりそれぞれ確信度として求める。

その後，欠陥発生特性と欠陥発生部位および超音波探傷試験結果より知識 Ut\_rules を用いて欠陥種類の推論を行い，さらに欠陥種類より知識 Figure\_rules を用いて欠陥形状を，欠陥形状と超音波探傷試験結果より知識 Estimation を用いて欠陥高さを推論する。

欠陥等級については，上記のフローとは別に，超音波探傷試験結果より知識 Defect\_class を用いて推論を行う。

(2) 推論方法 ここでは，欠陥種類の推定プロセスを用いて推論方法の説明を行う。

(a) 入力データの一致度<sup>(50)</sup> 入力データとして，「両側探傷した時のエコー高さの差は10dBより明らかに大きい」(B)，「色々なビーム路程でエコーが明らかに立ち上がる」(P)が与えられたとする。「両側探傷」については，図3・9(a)のメンバシップ関数により，「10dB以下である」(RS)との一致度は0.25，「10dBより大きい」(RB)との一致度は1.0となる。同様に，「色々なビーム路程でエコー」については，図3・9(d)のメンバシップ関数により，「立ち上がらない」(RN)との一致度は0.25，「立ち上がる」(RP)との一致度は1.0となる。

(b) ファジィ推論 以下に本システムで用いたルールの一例を示す。

(rule 密集ブローホール

(if ((欠陥発生部位は溶接金属部である))



((欠陥は密集ブローホールである))

(欠陥発生部位は溶接金属部である))

((両側探傷は10dB以下である))

((いろいろなビーム路程でエコーは立ち上がる))

(then ((欠陥は密集ブローホールである) 0.8))

このルールは欠陥種類として密集ブローホールを導くもので、条件部 (if部) は4つの条件節からなっている。結論部 (then部) の最後の数値はこのルール自身の確信度を表している。

このルールによりファジィ推論について説明する。ここでは、「欠陥発生部位は溶接金属部である」の確信度が0.7、欠陥発生特性として、「(欠陥は密集ブローホールである)(欠陥発生部位は溶接金属部である)」の確信度が1.0と推定されているものとする。

前述の入力データの一致度と合わせて考えると、上記のルールの各々の条件節の確信度は、「欠陥発生部位は溶接金属部である」は0.7、「(欠陥は密集ブローホールである)(欠陥発生部位は溶接金属部である)」は1.0、「両側探傷は10dB以下である」は0.25、「色々なビーム路程でエコーは立ち上がる」は1.0となる。条件部の確信度は、各々の条件節の確信度の最小値とし(論理積)、この例では0.25となる。

条件部が一致した場合に結論部が導かれる。そのとき、結論部の確信度は条件部の確信度とルール自身の確信度のうちの最小値をとるもの(論理積)とした。この例では、条件部の確信度0.25とルール自身の確信度0.8の最小値0.25が結論部の確信度となり、「欠陥は密集ブローホールである」の確信度は0.25が結論となる。

(c) 結論および出力 すべてのルールについてファジィ推論を行うことにより、各々の欠陥種類について確信度が求められ、確信度の最大値をとるものが可能性の最も高い欠陥種類となる。出力とし

ては、最初に入力データを、次に欠陥種類を確信度の大きい順に、さらに欠陥種類に応じた欠陥高さおよび欠陥等級を順次表示するようになっている。

(d) 推論過程の説明 欠陥種類および欠陥等級の推論において、その結果がどのように導かれたかを説明する機能を持っており、推論過程でたどってきたルールの連鎖を表示する。このとき本システムでは、それらのルールを全て記憶しているわけではなく、どのような連鎖になっているかだけをフレームに記憶させ、推論過程の説明を求められた時にのみ、その連鎖の記憶を用いてルールの連鎖を形成し表示するという方法を用いた。このことにより、ルールとルールとの連鎖関係がより分かりやすくなるという利点がある。

### 3・2・5 適用結果

ここで構築したファジィエキスパートシステムを実際の超音波探傷試験評価に適用した。本システムの推論結果の評価の妥当性を確かめるために、検査技術者による推定結果と比較したものを表3・2に示す。表には参考のために材料の開先形状および板厚を記した。本システムの推論結果として、確信度の大きい欠陥種類と、欠陥高さおよびJIS Z 3060による欠陥等級を記した。試験データ数は70例で、開先形状についてはV形が4例、レ(L)形が41例、X形が4例、K形が21例となっており、板厚は1例が75mmとなっている以外は8~45mmである。データに偏りがあるのはこれらのデータが実際の溶接構造物について行った試験から得られたものであることによる。欠陥種類は、ルート溶け込み不良(A)、内部溶け込み不良(B)、密集ブローホール(C)、単独ブローホール(D)、内部割れ(E)、開先面融合不良(FB)、層間融合不良(FM)、開先面スラグ巻き込み(GB)、層間スラグ巻き込み(GM)、熱影響部割れ(H)、ルート割れ(I)、妨害エコー(J)の12種類とした。ちなみに本



表3・2 本 シ ス テ ム の 適 用 結 果

No.	Profile of Groove	Thick-ness (mm)	Results by Experts (Type of Defect)	Results by This expert system		
				Type of Defect (Certainty factor)	Width of Defect (mm)	Defect class (JIS Z 3060)
1	L	19.0	F G	FM (1.00) GM (1.00)	0.19 0.19	4
2	V	16.0	F	FM (1.00) FB (0.44)	0.33 0.33	3
3	L	16.0	D F	D (1.00) FM (0.75) FB (0.75)	4.51 0.22 0.22	3
4	K	19.0	G F	GM (0.50) FM (0.50) GB (0.50) FB (0.50)	0.49 0.49 0.49 0.49	4
5	L	16.0	F	FB (1.00) FM (0.75)	0.51 0.51	4
6	L	16.0	J	FM (0.75) FB (0.75)	0.86 0.86	4
7	L	22.0	F	FM (0.75) FB (0.75) A (0.50)	0.34 0.34 0.34	4
8	L	18.0	C	C (1.00)	2.41	2
9	L	25.0	H	H (1.00)	0.71	4
10	L	9.0	D A	D (1.00) FM (0.50) FB (0.50)	3.47 0.20 0.20	2
11	L	16.0	F	FB (1.00) FM (0.94) D (0.75)	0.43 0.43 8.33	4
12	L	25.0	F	FM (0.75) FB (0.75)	0.33 0.33	4
13	V	21.0	F	FM (1.00)	0.30	3

14	L	12.0	D	D (1.00)	2.80	3
15	V	8.0	F	FB (1.00) FM (0.75) A (0.50)	0.17 0.17 0.17	3
16	K	25.0	F B	FM (1.00) B (0.75) FB (0.74)	0.23 0.23 0.23	4
17	L	28.0	C	C (1.00)	3.93	3
18	L	25.0	A	A (0.75) GB (0.46) FB (0.46)	0.43 0.43 0.43	4
19	K	28.0	B	B (1.00) FM (0.75)	0.33 0.33	4
20	K	28.0	F	FM (0.75)	0.24	4
21	K	32.0	A	A (0.89) FM (0.75) FB (0.75)	0.38 0.38 0.38	4
22	L	19.0	F	FM (0.75)	0.24	4
23	L	19.0	F	FM (0.75)	0.32	4
24	L	19.0	G F	GM (0.75) FM (0.50)	0.25 0.25	4
25	L	25.0	A	A (1.00) GB (0.50)	0.84 0.84	4
26	K	28.0	B	B (0.77) GM (0.75) GB (0.75)	0.31 0.31 0.31	4
27	L	30.0	G F	GM (1.00) FM (0.50)	0.36 0.36	1
28	L	20.0	F	FM (1.00)	0.12	4
29	L	19.0	F	FM (1.00) FB (0.89)	0.24 0.24	4
30	X	22.0	B	B (1.00) FM (0.75)	0.27 0.27	4



31	L	28.0	F	FM(0.75) FB(0.75)	0.31 0.31	4
32	K	36.0	B	B(0.85) GM(0.75) GB(0.75)	0.62 0.62 0.62	4
33	L	32.0	F	FM(1.00)	0.33	3
34	L	22.0	F	FM(0.75) FB(0.75)	0.26 0.26	4
35	K	36.0	B	B(0.93) FM(0.75) GB(0.75)	0.75 0.75 0.75	4
36	K	22.0	B	B(0.73) FM(0.50) FB(0.50)	0.61 0.61 0.61	4
37	K	25.0	F	FM(1.00) FB(0.41) C(0.40)	0.33 0.33 7.52	2
38	L	12.0	F	FM(0.75)	0.25	2
39	L	45.0	C G	C(1.00) GM(0.75) FM(0.75)	3.09 0.18 0.18	4
40	L	22.0	D G	D(1.00) GM(0.50) FM(0.50)	3.93 0.17 0.17	3
41	L	13.0	F	A(1.00) FB(0.75)	0.30 0.30	4
42	L	13.0	A	FM(1.00) FB(1.00) A(0.69)	0.33 0.33 0.33	4
43	L	12.0	F	FM(1.00) FB(1.00)	0.21 0.21	3
44	K	32.0	B	B(1.00) FM(0.50) FB(0.50)	0.67 0.67 0.67	4
45	L	16.0	F	FM(1.00) FB(1.00)	0.31 0.31	3

46	K	36.0	F	FB(1.00) FM(0.77)	0.29 0.29	1
47	L	16.0	A	FM(0.75) FB(0.75) A(0.59)	0.33 0.33 0.33	4
48	L	18.0	F	FM(1.00)	0.27	3
49	X	16.0	F	FM(0.50) FB(0.50)	0.13 0.13	1
50	L	17.0	F	FM(1.00) FB(0.59)	0.50 0.50	4
51	X	22.0	G F	GM(0.75) FM(0.56)	0.25 0.25	2
52	L	16.0	G F	GM(0.75) GB(0.56)	0.20 0.20	2
53	L	22.0	F	FM(1.00)	0.53	4
54	K	16.0	F	FM(1.00) FB(1.00)	0.17 0.17	4
55	K	21.0	G	GM(0.50) FM(0.50)	0.12 0.12	2
56	K	75.0	F	FB(0.75) B(0.62)	1.23 1.23	3
57	L	8.0	F	FM(1.00) FB(1.00) A(0.50)	0.25 0.25 0.25	3
58	V	18.0	A	A(0.62) FM(0.50) FB(0.50)	0.68 0.68 0.68	4
59	K	32.0	B F	B(1.00) FM(0.50) FB(0.50)	0.55 0.55 0.55	4
60	L	25.0	un- certain	GB(0.75) GM(0.50) FM(0.50)	0.54 0.54 0.54	2
61	K	22.0	F	FM(1.00) FB(0.73)	0.63 0.63	3



62	L	16.0	F	FM(1.00) FB(0.59)	0.31 0.31	4
63	L	16.0	F	FM(0.50) FB(0.50)	0.40 0.40	4
64	K	16.0	F	FB(1.00) FM(0.88)	0.52 0.52	4
65	L	12.0	F	FM(0.75) FB(0.75)	0.22 0.22	2
66	K	25.0	B	B(1.00) FM(0.50) FB(0.50)	0.56 0.56 0.56	4
67	K	32.0	F	B(1.00) FM(0.75) FB(0.75)	1.09 1.09 1.09	4
68	X	25.0	G F	GM(0.75) GB(0.75)	0.25 0.25	4
69	K	40.0	C G	C(0.94) H(0.40)	6.59 0.27	4
70	K	25.0	F	FM(1.00) FB(1.00)	0.34 0.34	3

where A : Lack of joint penetration at root zone  
 B : Lack of joint penetration at weld metal  
 C : Porosity  
 D : Blow hole  
 E : Crack at weld metal  
 F : Lack of fusion  
 FB : Lack of fusion at weld bond  
 FM : Lack of fusion at weld metal  
 G : Slag inclusion  
 GB : Slag inclusion at weld bond  
 GM : Slag inclusion at weld metal  
 H : Crack at heat affected zone  
 I : Crack at root zone  
 J : Parastic echo

システムでは、溶接金属部に発生するスラグ巻き込みおよび融合不良と、開先面に発生するスラグ巻き込みおよび融合不良とを区別し、それぞれ層間スラグ巻き込み、層間融合不良、開先面スラグ巻き込み、開先面融合不良としているが、検査技術者の推定結果ではその区別はされていない。

欠陥種類について、本システムの推論によって得られた結果のうち確信度が最大となった結果と検査技術者が推定した結果とが一致した割合は、70例中64例で約91%となり精度良く推論できていると考えられる。

以下に、本システムの推論結果と検査技術者の推定結果とが一致しなかったデータを中心にその原因を考察する。

(1) No. 6は、検査技術者の推定結果は妨害エコー(J)となっている。本システムでは、妨害エコーの判定については考慮していないので、このデータは本システムで全く推論できなかった唯一のものとなっている。

(2) No. 41, 67は、検査技術者の推定結果が融合不良(F)で、本システムの推論結果は1番目が溶け込み不良(A)に、2番目が融合不良(FB)になっている。逆にNo. 42, 47は検査技術者の推定結果は溶け込み不良(A)で、本システムの推論結果は1, 2番目が融合不良(FM, FB)に、3番目が溶け込み不良(A)になっている。欠陥種類の判定において、溶け込み不良と融合不良との判別は検査技術者でも難しく、また両者の有意差は比較的小さいとされている。このことから、これらのデータは検査技術者の推定結果と完全には一致していないが、検査技術者とほぼ同等の推論ができていると判断して差し支えないと考えられる。

(3) No. 55は、検査技術者の推定結果がスラグ巻き込み(G)で、本システムの推論結果は層間スラグ巻き込み(GM)と層間融合不良(FM)が同じとなっており結論が得られていないといえる。こ



の理由としては、推論時の条件部の確信度や結論の確信度を求める際に論理積を用いていることにありと考えられる。すなわち、条件部の確信度の値は、条件節の中の最小の確信度の値になるため、他の条件節の確信度の値には全く関係がなくなってしまうことにある。このことから、条件節の確信度の最小値が複数のルールで等しくなる場合、最小の確信度を持つ条件節以外の条件節の確信度がルールによって異なっているとしても、それらのルールから導かれた結論の確信度は等しくなり結論が得られなくなる可能性がある。このことは、今後演算方法を改良する余地が残されていることを示している。

### 3・2・6 おわりに

本節では、超音波探傷試験から溶接欠陥の推定を行う場合、その推定方法や得られるデータにあいまいさが含まれることに注目し、ファジィ理論や確信度を用いてそのあいまいさを扱えるようにすることにより精度の高いシステムを構築することができた。しかしながら、本章で問題になったのは以下の3点である。

(1) ファジィ推論では、ANDの演算に論理積が良く用いられており、本システムにおいても論理積を採用し良好な結果が得られた。しかしながら、一部のデータの推論結果に不適当な点がみられることから、本システムに適した演算方法（ファジィ結合演算子）の選択が必要と考えられる。

(2) 超音波探傷試験評価システムとしてさらに実用的なシステムとするために、妨害エコー等についても考慮できるようにルールの拡張が必要である。

(3) 本システムはワークステーション上のシステムとして構築したが、超音波探傷試験が現場で行われることが多いことを考えると、持ち運びが簡単なパソコンなどの上で動くシステムにする必要がある。

以上のことについては第4章と第5章に記述する。

## 第4章 ファジィ結合演算子の選定およびメンバシップ関数の改良

### 4・1 はじめに

第3章において超音波探傷試験評価のためのファジィエキスパートシステムのプロットタイプを構築し、実際の試験データについて適用し検査技術者の推定結果と比較した結果、90%を超える高い精度で一致することを確認した<sup>(18)</sup>。しかしながら、一部のデータにおいて不適当な結論が得られており、その理由としてファジィ結合演算子に論理積を用いていることが考えられたため、他の結合演算子についても検討する必要があることを指摘した。また、このシステムは一人の検査技術者の経験的知識に基づいて構築されており、検査技術者によって経験的知識や主観が異なることを考えると十分に普遍性を持っているとはいえない問題も残されている。

そこで、本章では超音波探傷試験評価システムに適したファジィ結合演算子の選定を行うと共に、複数の検査技術者から知識を獲得することにより、より普遍的なシステムを構築することを目的とする。

### 4・2 検査技術者による経験的知識およびあいまいさの相違

通常超音波探傷試験の結果から溶接欠陥の種類、欠陥寸法および欠陥等級の推定を行っているが、その中でも欠陥種類は欠陥寸法や合否判定の際に重要となるため、正確に推定することが要求される。欠陥種類は、欠陥の発生部位、エコーの波形およびエコー高さなどの情報から推定することができるが、精度良く判別し得る方法はいまだ完全には確立されていないのが現状と言える。したがって、検査技術者は各自の経験的知識に基づいて推定を行っている。そのため、その推定結果が検査技術者によって異なることも少なくない。ここでは、欠陥



種類の推定に用いる経験的知識が検査技術者によってどのように異なるかを複数の検査技術者に対して調査した結果に基づいて述べる。

欠陥種類を推定する際、その欠陥がルート部、溶接金属部、開先部、熱影響部もしくは母材部のどこに発生しているのかを把握することが重要となる。通常、欠陥の発生部位は探触子の位置、ビーム路程および屈折角から欠陥の位置を求め、溶接部の断面図にその位置を記入することにより推定している（図4・1）。しかしながら、溶接の施工精度や溶接の溶け込み量等を正確に把握することが困難なことおよび測定誤差等の問題から、欠陥発生部位を正確に推定することは困難である。特に、欠陥が開先面付近にある場合、その欠陥が溶接金属部もしくは開先部のどちらに位置するかを判別することは難しく（図4・1）、検査技術者の主観による違いが現れやすい。

また、エコーの波形によって融合不良、スラグ巻き込みおよび密集ブローホールの判別を行うことができるが、その判定基準が検査技術者によって異なっている。例えば、ある検査技術者はエコーの波形がシャープに立ち上がる場合は融合不良、ギザギザに立ち上がる場合はスラグ巻き込み、色々なビーム路程で立ち上がる場合は密集ブローホールと推定するのに対して、他のある検査技術者はエコーの波形がシャープに立ち上がる場合およびギザギザに立ち上がる場合は共に融合不良、色々なビーム路程で立ち上がる場合はスラグ巻き込みもしくは密集ブローホールと推定している。

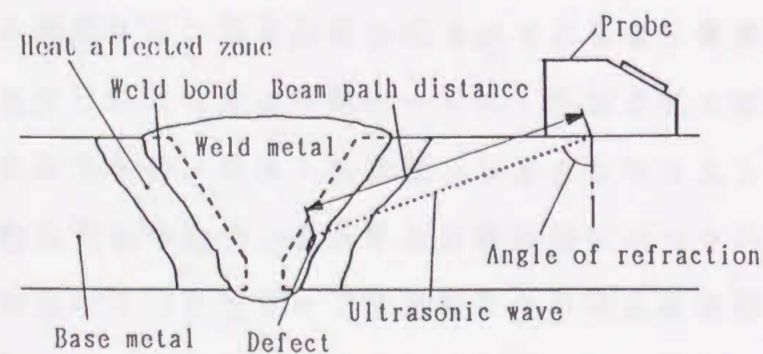


図4・1 超音波探傷試験（斜角探傷法）

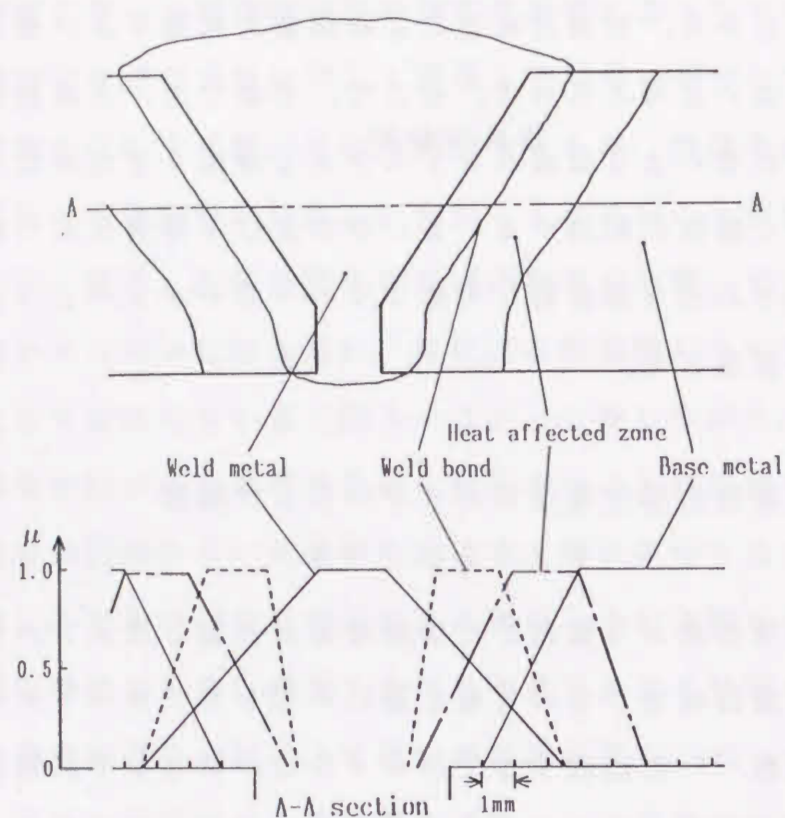
これらのことから、検査技術者による経験的知識やあいまいさの違いは無視できないと考えられる。そこで、本章では、各検査技術者の主観に左右されないより普遍的なシステムを構築するために、複数の検査技術者から経験的知識やあいまいさおよび不確実さについての情報を獲得し、それらを総合的に考慮してルールやメンバシップ関数および確信度を決定した。

#### 4・3 欠陥発生部位推定用のメンバシップ関数

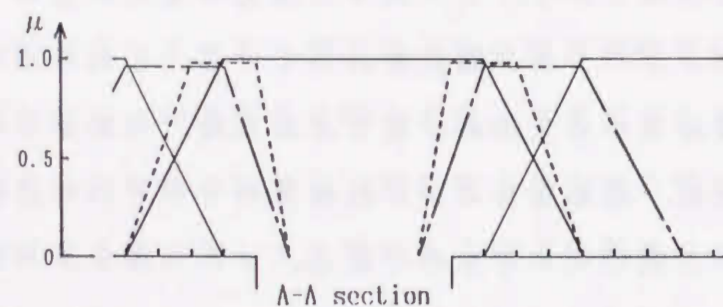
複数の検査技術者から獲得した情報を基に改良したメンバシップ関数と1人の検査技術者からの情報を基に作成したプロットタイプのメンバシップ関数<sup>(18)</sup>を比較する。図4・2に、例として欠陥種類を推定する上で最も重要となる欠陥発生部位を推定するためのメンバシップ関数を比較したものを示す。これはV形開先の場合のもので、断面図のA-A線上のメンバシップ関数を表している。これらのメンバシップ関数は、試験結果から欠陥の位置を求めた後、欠陥がルート部、溶接金属部、開先部、熱影響部および母材部のそれぞれの部位に位置する確信度を求める際に用いるものである。これらのメンバシップ関数は、複数の検査技術者から獲得した情報を基に作成し、さらに各検査技術者の推定結果に最も良く一致するようにチューニングを行ったものである。他のメンバシップ関数および確信度についても同様に決定した。

この図より、開先部のメンバシップ関数はほぼ同じであるが、溶接金属部および熱影響部のそれは開先面付近において改良後の方がプロットタイプよりその値が小さくなっていることがわかる。これによって、プロットタイプに比較して、改良後の方は各部位のメンバシップ関数の確信度が1となる部分の重複する範囲が減少している。これは、複数の検査技術者から獲得した多くの情報が利用できたことから、よ





(a) 改良後



(b) プロットタイプ

図4・2 欠陥発生部位推定用メンバシップ関数（V形開先）

り実際に即したメンバシップ関数が作成できたためと考えられる。

#### 4・4 推論方法

ここでファジィ理論による推論方法について述べる。以下に、データおよびルールを示す。

$((x_1 \text{ is } P_1) u_1)$  : データ1

$((x_2 \text{ is } P_2) u_2)$  : データ2

.....

$((x_m \text{ is } P_m) u_m)$  : データm

(rule i rule-name

(if : 条件部

$((x_1 \text{ is } Q_1))$  : 条件節1

$((x_2 \text{ is } Q_2))$  : 条件節2

.....

$((x_n \text{ is } Q_n))$  : 条件節n )

(then  $((y_i \text{ is } R_i) v_i))$  : 結論部 )

ここで、 $x_i$ は検査項目、 $y_i$ は仮説レベル、 $P_i$ 、 $Q_i$ および $R_i$ は検査項目 $x_i$ および仮説レベル $y_i$ に対するファジィ集合（メンバシップ関数）で定義された評価、 $u_i$ および $v_i$ はそれぞれデータおよびルール自身の信ぴょう性（確からしさ）を表す確信度である。

まず、ルールベースから最初のルールを取り出す。その後ルールの条件部の各条件節と各データの一致度 $m_{xi}$ を求める。 $m_{xi}$ は、 $P_i$ および $Q_i$ のメンバシップ関数の交わりの高さとして求められる。次に、各条件節の確信度 $m_i$ を $m_{xi}$ と $u_i$ の最小値（論理積）として求める。さらに、条件部全体の確信度 $m$ は各条件節の確信度 $m_i$ の最小値（論理積）となる。条件部全体の確信度が求めれば、それとルール自身の確信度の最小値（論理積）が結論部の確信度とみなされ、それを結論とする。最初のルールについての推論が終了すると、ルールベースから次のルールを取り出し、そのルールについても推論を行う。この操作をすべてのルールに対して行うことにより各結論部に対する確信度が求められる。



上記の推論方法で示したように、ファジィ推論ではファジィ集合の結合演算子として論理積が良く用いられる。しかし、論理積ではファジィ集合間の相互作用が考慮されていないために、例えば条件部全体の確信度を求める場合では、最小の確信度をとる条件節以外の条件節の確信度が変化しても条件部全体の確信度は不変となり不合理な結論を導く可能性がある。プロットタイプにおいて、一部のデータに不適当な結果が現れたのは、これが原因と考えられる。したがって、他の演算子についても検討し、このシステムに適した結合演算子を選定する必要があると考えられる。

#### 4・5 ファジィ結合演算子

4・5・1 種々のファジィ結合演算子 ファジィ集合の結合演算については、 $t\text{-norm}$ 、 $t\text{-conorm}$ <sup>(56)</sup>、 $\gamma$ 演算<sup>(57)</sup>、拡張 $\gamma$ 演算<sup>(58)</sup>あるいは学習型ファジィ結合演算子<sup>(59)</sup>などが提案されている。これらの結合演算子は、意志決定問題における意志決定者の選好構造の表現およびファジィ検索における結合演算子として用いられ、良好な成果が挙げられている。ここでは、基本的な結合演算子である $t\text{-norm}$ 、 $t\text{-conorm}$ および平均演算子について説明する。

$t\text{-norm } T$ は、 $T(x, y): [0, 1] \times [0, 1] \rightarrow [0, 1]$ なる関数で積演算を表しており、以下の条件を満たす。

- ①  $T(x, 1) = x, T(0, x) = 0$  (境界条件)
- ②  $x_1 \leq x_2, y_1 \leq y_2$  (単調性)  
 $\rightarrow T(x_1, y_1) \leq T(x_2, y_2)$
- ③  $T(x, y) = T(y, x)$  (交換性)
- ④  $T(x, T(y, z)) = T(T(x, y), z)$  (結合性)

$t\text{-conorm } S$ は、 $t\text{-norm } T$ に双対な演算で和演算を表しており、以下の関係がある。

$$S(x, y) = 1 - T(1 - x, 1 - y)$$

代表的な $t\text{-norm } T$ および $t\text{-conorm } S$ として以下の演算子がある。

##### I $t\text{-norm } T$ (積演算)

$$\text{論理積: } x \wedge y = \min\{x, y\} \quad \dots\dots\dots(1)$$

$$\text{代数積: } x \cdot y = x y \quad \dots\dots\dots(2)$$

$$\text{限界積: } x \odot y = 0 \vee (x + y - 1) \quad \dots\dots\dots(3)$$

$$\text{激烈積: } x \triangle y = \begin{cases} x & (y = 1) \\ y & (x = 1) \\ 0 & (x, y < 1) \end{cases} \quad \dots\dots\dots(4)$$

##### II $t\text{-conorm } S$ (和演算)

$$\text{論理和: } x \vee y = \max\{x, y\} \quad \dots\dots\dots(5)$$

$$\text{代数和: } x \dot{+} y = x + y - x y \quad \dots\dots\dots(6)$$

$$\text{限界和: } x \oplus y = 1 \wedge (x + y) \quad \dots\dots\dots(7)$$

$$\text{激烈和: } x \vee y = \begin{cases} x & (y = 0) \\ y & (x = 0) \\ 1 & (x, y > 0) \end{cases} \quad \dots\dots\dots(8)$$

また、 $t\text{-norm } T$ 、 $t\text{-conorm } S$ には種々のパラメータ $P$ をもつ演算子が提案されており、パラメータ $P$ を変えることにより任意の $t\text{-norm}$ もしくは $t\text{-conorm}$ を表現することができる。

さらに、平均演算子として算術平均(AM)、幾何平均(GM)、双対な幾何平均(CGM)などがある。これらの演算子の間には次のような大小関係がある。

$$\wedge \leq \odot \leq \cdot \leq \triangle \leq GM \leq AM \leq CGM \leq \vee \leq \dot{+} \leq \oplus \leq \vee$$

4・5・2 本システムで用いたファジィ結合演算子 本システムのルールは複数の条件節からなっており、これらの条件節



はすべてANDで結合されている。したがって、本システムのファジィ結合演算子としてはt-normを用いるのが適切と考えられる。

また、ファジィ集合間の依存関係によって、適切なファジィ結合演算子が変わる<sup>(60)</sup>。例えば、積演算の場合、ファジィ集合間に正の依存関係がある場合は論理積、負の依存関係がある場合は限界積、依存関係がなく独立な場合には代数積を用いると適切な値が得られる。また、論理積から限界積の間で得られる集合はAND演算がとり得る値の範囲を完全に含んでいる<sup>(60)</sup>。

以上のことから、ファジィ結合演算子として論理積、代数積、限界積および激烈積を当てはめ、本システムに適した結合演算子の選定を行うことにした。ここで、ファジィ結合演算子として論理積、代数積、限界積および激烈積を用いた場合の推論結果の違いについて具体的な例を用いて説明する。以下に本システムで用いたルールの一例を示す。

```
(rule 単独ブローホール
  (if ((欠陥発生部位 は 溶接金属部である))
      ((振子走査 は 小さい)))
  (then ((欠陥 は 単独ブローホールである) 0.8)))
```

このルールは欠陥種類として単独ブローホールを導くもので、条件部(if部)は2つの条件節からなっている。また、結論部(then部)の最後の数値は、このルール自身の確からしさを表す確信度である。ここで、以下のデータが与えられているものとする。

((欠陥発生部位 は 溶接金属部である) 0.8)

((振子走査 は 約8dBである) 1.0)

上記のデータはそれぞれ「欠陥発生部位 は 溶接金属部である」の確信度が0.8、「振子走査 は 約8dBである」の確信度が1.0であることを表している。ルールおよびデータ中の「振子走査」は、

「振子角 $\pm 15^\circ$ で振子走査した時のエコーの低下」を意味している。

まず、これらのデータと上記のルールの条件部の各条件節との一致度を求める。「欠陥発生部位 は 溶接金属部である」についてはデータと条件節が全く同一なのでその一致度は1.0となる。「振子走査」については「小さい」と「約8dBである」のそれぞれのメンバシップ関数が図4・3のように与えられているので、その一致度は0.4となる。

次に、各条件節の確信度を求める。これは各条件節とデータとの一致度とデータの確信度との積で求められる。したがって、「欠陥発生部位 は 溶接金属部である」の確信度は0.8( $=1.0 \times 0.8$ )、「振子走査 は 小さい」の確信度は0.4( $=0.4 \times 1.0$ )となる。

さらに、これらの各条件節の確信度から条件部全体の確信度を求めるわけであるが、条件部全体の確信度はファジィ結合演算子によって異なり、以下のように求まる。

論理積( $\wedge$ ):  $\min\{0.8, 0.4\} = 0.4$

代数積( $\cdot$ ):  $0.8 \times 0.4 = 0.32$

限界積( $\odot$ ):  $0.8 + 0.4 - 1.0 = 0.2$

激烈積( $\wedge$ ):  $0.8 \wedge 0.4 = 0.0$

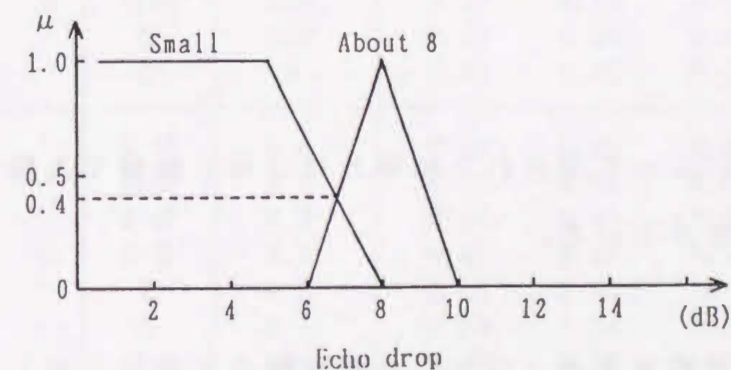


図4・3 振子走査判定用メンバシップ関数



最後に、条件部全体の確信度とルール自身の確信度との積が結論の確信度となる。したがって、「欠陥種類は単独ブローホールである」の確信度は、論理積では0.32(=0.4×0.8)、代数積では0.256(=0.32×0.8)、限界積では0.16(=0.2×0.8)、激烈積では0.0(=0.0×0.8)となる。

#### 4・6 推 論 結 果

本システムに適したファジィ結合演算子を選定するために、結合演算子として論理積、代数積、限界積および激烈積を当てはめ、20例の試験データについて推論を行い、検査技術者の推定結果と比較した。表4・1にその比較したものを示す。ここで、検査技術者によって推定結果が異なるため、検査技術者らに検討調整してもらい、最終的に一本化して得られたものを検査技術者の推定結果とした。

4・6・1 ファジィ結合演算子の選定 まず、データNo. 1を例にとり、本システムの推論結果が演算子によってどのように違うか以下に述べる。

##### 論理積(∧)

開先面融合不良(FB)、開先面スラグ巻き込み(GB)、層間融合不良(FM)および層間スラグ巻き込み(GM)の4つの欠陥種類の確信度が同じ値(0.67)となり、結論が得られていない。

##### 代数積(・)

検査技術者によって得られた結果とほぼ同じ順番で欠陥種類の確信度が推論できている。

##### 限界積(⊙)

開先面スラグ巻き込み(GB)と層間融合不良(FM)の確信度の順番が検査技術者の結果と逆になっている。

表4・1 検査技術者の推定結果とシステムの推論結果との比較

Data No.	Type of Defect	Results by Experts	Results by System			
			∧	・	⊙	△
1	FB	0.9	0.67	0.67	0.67	0.67
	GB	0.5	0.67	0.50	0.25	0
	FM	0.3	0.67	0.45	0.34	0
	GM	0.2	0.67	0.34	0.09	0
	C	0.1	0.17	0.03	0	0
	D	0	0.25	0.07	0	0
	H	0	0.08	0.08	0	0
2	FB	0.9	0.82	0.82	0.82	0.82
	GB	0.5	0.75	0.62	0.57	0
	FM	0.3	0.74	0.74	0.74	0.74
	GM	0.2	0.74	0.55	0.49	0
	C	0.1	0.17	0.03	0	0
	D	0	0.25	0.06	0	0
	E	0	0.08	0.06	0	0
3	FB	0.7	0.67	0.67	0.67	0.67
	GB	0.5	0.67	0.63	0.59	0
	FM	0.4	0.67	0.45	0.34	0
	GM	0.3	0.67	0.43	0.26	0
	H	0.2	0.08	0.08	0.08	0.08
	E	0.1	0.20	0.20	0.20	0.20
	C	0.1	0.22	0.15	0	0
4	C	0.9	1.00	1.00	1.00	1.00
	GM	0.3	0.50	0.25	0	0
	FM	0.2	0.20	0.10	0	0
	D	0.1	0	0	0	0
5	FM	0.9	0.83	0.83	0.83	0.83
	GM	0.8	0.83	0.78	0.77	0
	E	0.2	0.30	0.30	0.30	0.30
	C	0	0.22	0.22	0.22	0.22
6	FM	0.8	0.81	0.81	0.81	0.81
	GM	0.5	0.75	0.61	0.56	0
	FB	0.3	0.41	0.41	0.41	0.41
	GB	0.2	0.41	0.31	0.17	0
	E	0.2	0.15	0.12	0	0
	C	0.1	0.24	0.14	0	0
	D	0	0.40	0.16	0	0



7	B	0.9	1.00	1.00	1.00	1.00
	FB	0.5	0.83	0.78	0.77	0
	FM	0.5	0.72	0.60	0.55	0
	GB	0.3	0.75	0.59	0.52	0
	GM	0.2	0.72	0.45	0.30	0
	E	0.2	0.22	0.22	0.22	0.22
	C	0.1	0.17	0.13	0	0
	H	0.1	0.05	0.05	0.05	0.05
8	B	1.0	1.00	1.00	1.00	1.00
	GB	0.7	0.83	0.78	0.77	0
	FB	0.6	0.83	0.83	0.83	0.83
	GM	0.4	0.37	0.29	0.03	0
	FM	0.4	0.37	0.31	0.20	0
	H	0.2	0.24	0.24	0.24	0.24
	E	0.1	0.11	0.11	0.11	0.11
	C	0	0.22	0.08	0	0
9	B	0.8	0.91	0.91	0.91	0.91
	GB	0.7	0.83	0.71	0.68	0
	FB	0.4	0.83	0.76	0.75	0
	GM	0.3	0.33	0.26	0.10	0
	FM	0.3	0.33	0.28	0.16	0
	H	0.3	0.27	0.27	0.27	0.27
	E	0.1	0.10	0.10	0.10	0.10
	C	0	0.22	0.07	0	0
10	GM	0.8	0.93	0.87	0.87	0
	FM	0.7	0.93	0.93	0.93	0.93
	E	0.1	0.08	0.07	0	0
	C	0	0.24	0.21	0.12	0
	D	0	0.25	0.07	0	0
11	FB	0.8	0.83	0.80	0.79	0
	GB	0.7	0.83	0.75	0.73	0
	FM	0.5	0.72	0.60	0.55	0
	GM	0.5	0.72	0.56	0.49	0
	C	0.1	0.24	0.23	0.22	0
	D	0.1	0.25	0.07	0	0
	H	0	0.06	0.06	0.06	0.06
12	FM	0.7	1.00	1.00	1.00	1.00
	GM	0.4	0.75	0.75	0.75	0.75
	E	0.2	0.15	0.15	0.15	0.15
	C	0.1	0.24	0.17	0.17	0
	D	0	0.40	0.20	0	0

13	B	0.8	0.76	0.76	0.76	0.76
	FB	0.4	0.67	0.56	0.51	0
	GB	0.3	0.67	0.42	0.26	0
	FM	0.3	0.31	0.21	0	0
	H	0.2	0.28	0.28	0.28	0.28
	GM	0.2	0.31	0.16	0	0
	E	0	0.09	0.09	0.09	0.09
	C	0	0.17	0.05	0.01	0
14	FM	0.9	0.79	0.79	0.79	0.79
	FB	0.5	0.48	0.48	0.48	0.48
	GM	0.3	0.75	0.59	0.54	0
	GB	0.3	0.48	0.36	0.23	0
	E	0.1	0.15	0.12	0.09	0
	H	0	0.01	0.01	0.01	0.01
	C	0	0.24	0.14	0	0
	D	0	0.40	0.16	0	0
15	FM	0.8	0.83	0.83	0.83	0.83
	GM	0.7	0.83	0.78	0.77	0
	C	0.2	0.24	0.24	0.24	0.24
	D	0	0.25	0.10	0	0
16	A	1.0	1.00	1.00	1.00	1.00
	I	0.2	0.30	0.30	0.30	0.30
	H	0.2	0.08	0.08	0.08	0.08
	FB	0.1	0.36	0.24	0.03	0
	C	0	0.22	0.22	0.22	0.22
	FM	0	0.36	0.24	0.03	0
	GB	0	0.36	0.23	0	0
	GM	0	0.36	0.23	0	0
17	E	0	0.11	0.11	0.11	0.11
	FM	1.0	1.00	1.00	1.00	1.00
	GM	0.5	0.75	0.75	0.75	0.75
	E	0.1	0.15	0.15	0.15	0.15
	D	0	0.40	0.20	0	0
18	C	0	0.24	0.17	0.17	0
	FB	0.8	0.63	0.52	0.46	0
	GB	0.2	0.63	0.39	0.21	0
	B	0.2	0.30	0.30	0.30	0.30
	FM	0.2	0.23	0.19	0.06	0
	GM	0.2	0.23	0.14	0	0
	H	0.1	0.30	0.30	0.30	0.30
	E	0	0.07	0.07	0.07	0.07
	C	0	0.17	0.04	0	0



19	GB	0.7	0.83	0.78	0.77	0
	FB	0.6	0.83	0.83	0.83	0.83
	FM	0.6	0.64	0.53	0.47	0
	GM	0.5	0.64	0.50	0.41	0
	B	0.2	0.25	0.24	0.22	0
	E	0.2	0.19	0.19	0.19	0.19
	H	0	0.10	0.10	0.10	0.10
	C	0	0.22	0.04	0	0
20	C	0.8	1.00	1.00	1.00	1.00
	GB	0.3	0.50	0.50	0.50	0.50
	GM	0.2	0.43	0.22	0	0
	H	0.2	0.21	0.21	0.21	0.21
	FB	0.2	0.20	0.20	0.20	0.20
	FM	0.2	0.21	0.09	0	0
	B	0.1	0.50	0.38	0.25	0
	E	0.1	0.13	0.05	0	0

where A : Lack of joint penetration at root zone  
 B : Lack of joint penetration at weld metal  
 C : Porosity  
 D : Blow hole  
 E : Crack at weld metal  
 FB : Lack of fusion at weld bond  
 FM : Lack of fusion at weld metal  
 GB : Slag inclusion at weld bond  
 GM : Slag inclusion at weld metal  
 H : Crack at heat affected zone  
 I : Crack at root zone

激烈積 ( $\wedge$ )

開先面融合不良 (FB) 以外の欠陥種類の確信度がすべて0となっている。

ここで、論理積による推定結果において確信度が同じである欠陥種類が4つ現れている原因は、前述したように論理積ではファジィ集合間の相互作用が考慮されないためと考えられる。以上の結果から、代数積を用いた場合が最も有効であることがわかる。

次に、全データについて検査技術者によって推定された欠陥種類と

本システムによって推定された欠陥種類の一致度を調べるために、両者の結果において、確信度が最も高い欠陥種類が一致したデータ数およびその割合と確信度が1番目と2番目の欠陥種類が共に一致したデータ数およびその割合を表4・2に示す。この結果から、演算子として代数積を用いた場合の推論結果が、検査技術者の推定結果に最も良く一致していることがわかる。

以上のことから、本システムのファジィ結合演算子として、代数積が最も適していると考えられる。

4・6・2 本システムと検査技術者による推定結果の比較  
 ここでは、本システムのファジィ結合演算子として代数積を用いた場合の推論結果と検査技術者による推定結果を比較する。No. 1, 3, 4, 5, 6, 7, 11, 12, 13, 15, 17, 18および20の13個のデータについては、本システムと検査技術者の推定結果は良く一致している。以下に、それ以外のデータについて一致しなかった理由を考察する。

データNo. 2および14は、確信度が2番目と3番目の欠陥種類が、本システムと検査技術者において逆に推定されている。一般に、欠陥が開先面付近に位置している時、その欠陥が溶接金属部もしくは開先部のどちらに発生しているのかを正確に判定することは非常に難しいとされている。また、スラグ巻き込みと融合不良との判別は、普

表4・2 各結合演算子を用いた場合のシステムの推論結果と検査技術者の推定結果の一致度(全データ数20)

Fuzzy connective	First	First and Second
Logical product ( $\wedge$ )	12 (60%)	6 (30%)
Algebraic product ( $\cdot$ )	18 (90%)	14 (70%)
Bounded product ( $\odot$ )	18 (90%)	12 (60%)
Drastic product ( $\wedge$ )	16 (80%)	5 (25%)



通エコーの波形の違いによって行っているが、このエコーの波形の判定にはあいまいな面が多いので、検査技術者でも熟練が必要とされている。上記の2つのデータにはこれらの問題が同時に存在しており、非常に判定の難しい試験データと言える。

データNo. 8, 9, 10および19では、スラグ巻き込み (GB, GM) と融合不良 (FB, FM) の判定に違いがみられる。しかしながら、スラグ巻き込みと融合不良の判定は上記のように難しいとされていることから、No. 8, 10, 19については、検査技術者と同程度の推論ができていると考えられる。

またNo. 16については、検査技術者は密集ブローホール (C), 層間融合不良 (FM), 開先面スラグ巻き込み (GB), 層間スラグ巻き込み (GM) および内部割れ (E) である可能性はないと推定しているのに対して、システムはこれらの欠陥種類の確信度を0.11~0.24と推論している。しかし、このデータについては欠陥種類としてルート溶け込み不良の可能性が非常に高いことが明らかであるので、これらの違いは特に問題はないものと考えられる。

以上のことから、本システムは、No. 2, 9および14以外のデータについては、検査技術者と同程度の推論ができていると判断できる。また、この3つのデータについても、確信度が最も高い欠陥種類については検査技術者の推定結果と一致している。したがって、本システムは十分実用に耐え得るものであると判断して差し支えないものと考えられる。

#### 4・7 おわりに

本章では、より普遍的な超音波探傷試験評価システムを構築するために、複数の検査技術者から知識を獲得することにより、ルール、メンバーシップ関数および確信度を決定した。さらに、本システムにおい

て種々のファジィ結合演算子を用いて得られた推論結果と検査技術者による推定結果とを比較した結果、本システムの結合演算子としては代数積が最も有効であることがわかった。また、その推論結果は検査技術者による推定結果と良い一致を示し、精度の高いシステムであることが確認された。



## 第5章 パソコン用システムの構築および

### 妨害エコー判定用知識の補充

#### 5・1 はじめに

超音波探傷試験は現場で行われることが多いため、システムは持ち運びが可能なものが望ましいが、開発したシステムはワークステーション上に構築しており、推論速度の点では問題ないが、実用面では十分とはいえないと考えられる。したがって、パソコン上で動くシステムを構築することは、実用上有益と考えられる。

これまでに開発したシステムについては第3章および第4章で述べたが、それらのシステムは妨害エコーについて全く考慮されていないことなどから改良の余地が残されている。

そこで本章では、より実用性を高めるためにパソコン上で使用できるシステムの構築を行うと共に妨害エコーについても判定できるように知識の補充を行う。

#### 5・2 概要

本システムは推論エンジン、知識ベースおよびワーキングメモリから構成されている。通常、知識やデータの量が多くなると推論に要する時間が長くなるので、この問題を解消するために知識やデータをグループ化する黑板モデル<sup>(31)</sup>を採用している。知識はプロダクションルール形式とし、フレームモデルによってグループ化する。また、パソコンはNEC PC-9801DXを用い、言語はm u L i s p-86(MS-DOS上で稼動)を使用する。

5・2・1 パソコン用システム 超音波探傷試験は現場で行い、その場で合否を判定することが多いため、持ち運びが可能な超音波探

傷試験評価システムを構築することは実用性の面で大きな意味があると考えられる。そのためには、この評価システムをパソコン上のシステムとして移植する必要がある。パソコン用のシステムを構築する際、ワークステーションの言語と若干の違いがあることの他に、推論速度が遅いという問題がある。そこで、パソコン用のシステムを構築する際に、推論速度の短縮についても検討する必要がある。

推論時間を短縮するためには、推論を効率化することが考えられ、その方法の一つとして黑板モデルがある。このモデルは、推論の流れを階層化し、階層毎にグループ化された知識が結び付けられており、個々の推論に必要な知識のみを利用することで効率的な推論を行うものである。しかし、推論の階層化および知識のグループ化が不十分な場合は効率化も不十分となる。したがって、各知識がどのような役割を分担し、どのような状況で利用されるかを十分に把握し体系化する必要がある。本システムにおいても効率的な推論を行うために黑板モデルを使用していることから、知識の体系化を再検討し、不十分である部分についてはモデルを再構築し、より効率的な推論が行えるように変更する。

5・2・2 妨害エコー 検査技術者が超音波探傷試験を行う場合、はじめにCRT上のエコーが欠陥によるものなのか欠陥以外の妨害エコーなのかを判定した後、欠陥によるエコーについてのみ判定を行い溶接の合否を決定している。妨害エコーは溶接の余盛りや裏当金などの溶接部の形状によって発生するため、エコーが欠陥によるエコーか妨害エコーかを判定することは非常に難しく熟練を要する。しかし、妨害エコーは欠陥とは無関係なエコーであるので、この判定を正確に行えるかどうか超音波探傷試験評価の中でも最も重要な問題である。これまでに構築したシステム<sup>(18)・(19)</sup>では、エコーは欠陥によるものとして判定を行っており、妨害エコーについては全く考慮されていない。しかしながら、このシステムは初心者でも十分に扱うこ



とができることを前提としていることから、妨害エコーについても判定できるように知識の補充を行う。

### 5・3 推論時間の短縮

5・3・1 黑板モデルの改良 図5・1にこれまでに構築したシステムの黑板モデルを示す。この図に基づいて推論手順を述べる。

まず、溶接条件および超音波探傷試験結果のデータを対話式に入力する。入力が終わると、溶接条件データより知識 Weld\_rules を用いて欠陥発生特性を、超音波探傷試験結果のデータより知識 Data\_t を用いて欠陥発生部位を推論する。欠陥発生特性は、材質、溶接方法、開先形状などの違いによる各々の欠陥の発生頻度を表している。その後、欠陥発生特性と欠陥発生部位および超音波探傷試験結果より知識 Ut\_rules を用いて欠陥種類の推論を行い、さらに欠陥種類より知識 Figure\_rules を用いて欠陥形状を、欠陥形状と超音波探傷試験結果より知識 Estimation を用いて欠陥高さを推論する。欠陥等級については超音波探傷試験結果より知識 Defect\_class を用いて推論を行う。

この黑板モデルでは、6つの階層化された推論に対して6つのグループ化された知識ソースが作用するようになっている。この知識ソースのうち Weld\_rules はルール数が188個と全ルール数269個の大部分を占めており、知識の体系化が不十分であることが考えられた。

そこで、推論全体の流れを見直した結果、Weld\_rulesは4つの小グループに分ける方が合理的であると判断し、本システムの黑板モデルの改良を行った。

5・3・2 推論時間の変化 これまでに構築したシステムと黑板モデル改良後のシステムの推論時間の違いを10個のデータについて比較した結果、改良後のシステムの推論時間は以前のシステムより平均約24%減少することが確認された。

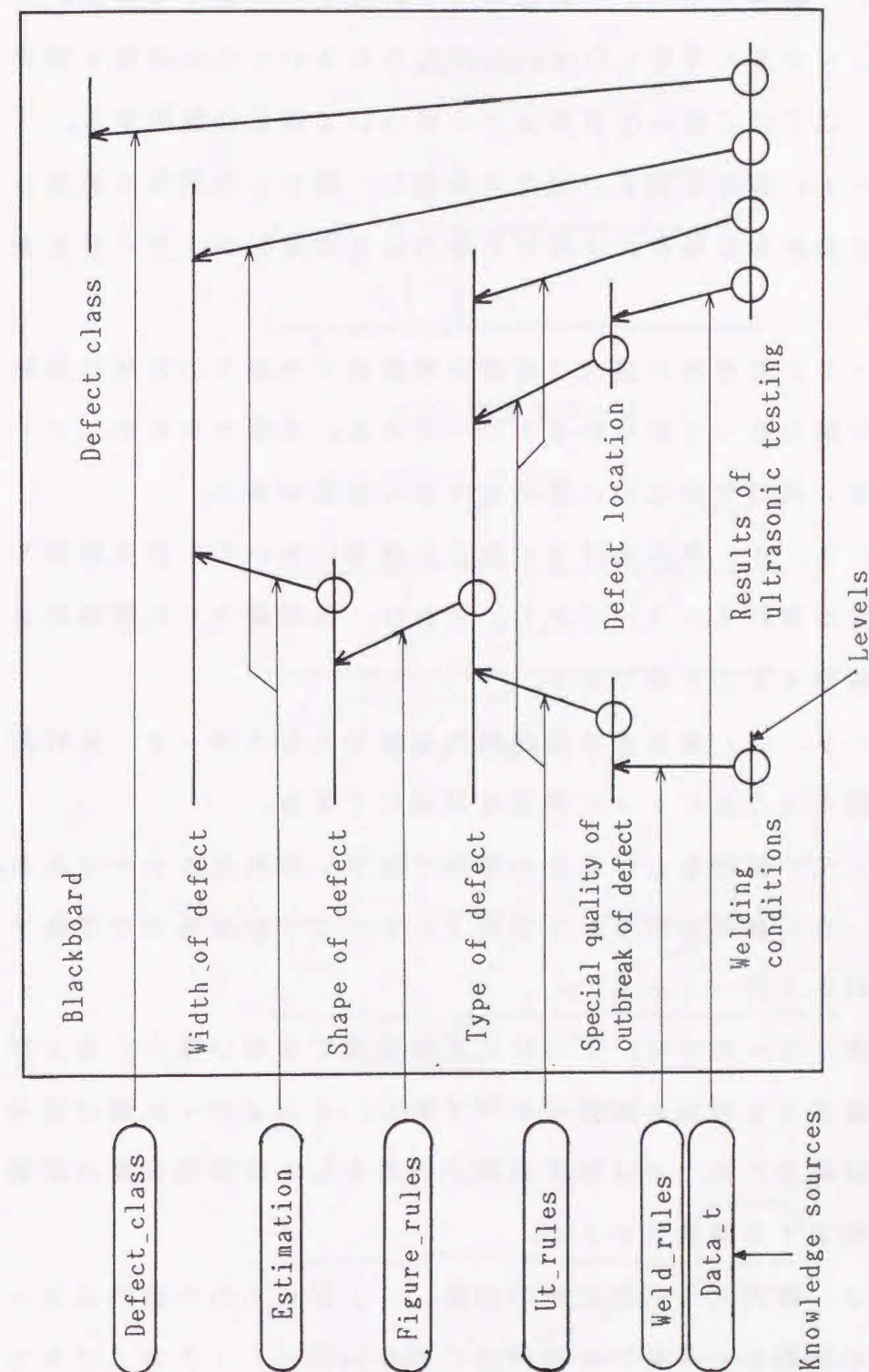


図5・1 黑板モデル



## 5・4 妨害エコーの判定

5・4・1 妨害エコー 妨害エコーの発生には様々な状況が考えられるが、そのうち発生しやすいと考えられるいくつかの例を図5・2に示す。以下にこれらの妨害エコーについて簡単に説明する。

妨害エコー1：突合せ継手の場合に発生し、特にレ形開先に発生しやすく、裏当金がある場合には母材と裏当金との隙間が大きいほど発生しやすい。

妨害エコー2：超音波が突合せ溶接の裏波部で余盛りの方向に反射し、余盛りで跳ね返って得られるエコーである。余盛りからのエコーは比較的大きく母材欠陥などに誤判定する可能性がある。

妨害エコー3，4：裏当金付きの突合せ継手において1回反射法で探傷する場合に現れるエコーである。これは、反対側からの探傷によって容易に判別することができる。

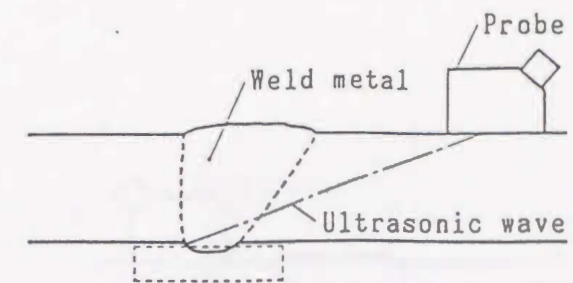
妨害エコー5，6：裏当金の補強隅肉溶接からのエコーで、反射源の位置を確認することによって判別は容易にできる。

妨害エコー7：裏当金と母材との隙間で反射し現れたエコーである。

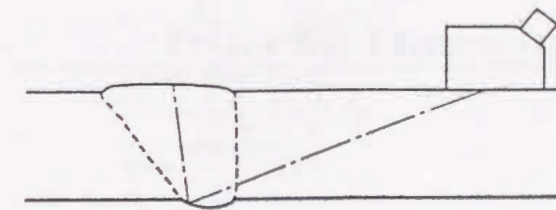
妨害エコー8：裏当金付きのT字継手において1回反射法で探傷する場合に現れるエコーである。

以上の妨害エコーの中で、1（特に両側探傷できない場合）および7，8は熟練者でも判別の困難なものであり、1および7は溶け込み不良もしくは融合不良，8は開先面融合不良もしくは溶接金属内部割れなどと誤判定する可能性がある。

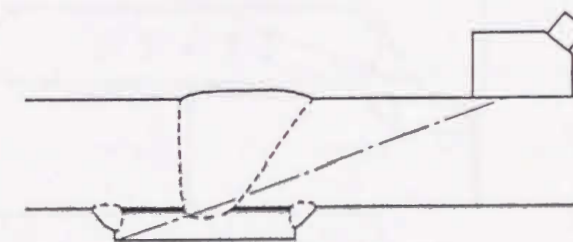
5・4・2 妨害エコー判定用の知識 上記の8つの妨害エコーのうち判別の困難な1（特に両側探傷できない場合），7および8の3つの妨害エコーの判別法について説明する。図5・3～5・5に、これらの妨害エコーを判定するためのフローチャート<sup>(55)</sup>を示す。以



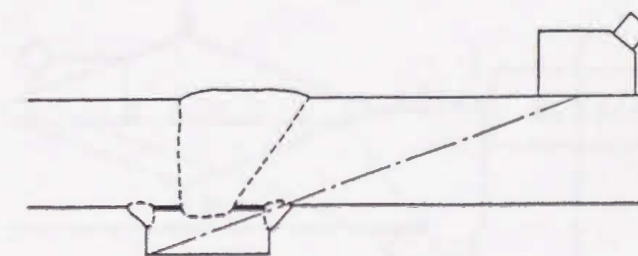
(a) 妨害エコー1



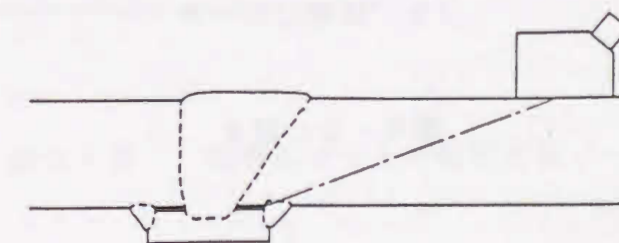
(b) 妨害エコー2



(c) 妨害エコー3



(d) 妨害エコー4



(e) 妨害エコー5

図5・2 妨害エコーの例



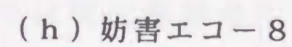
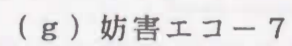
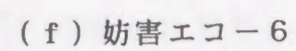


図 5・2 の続き

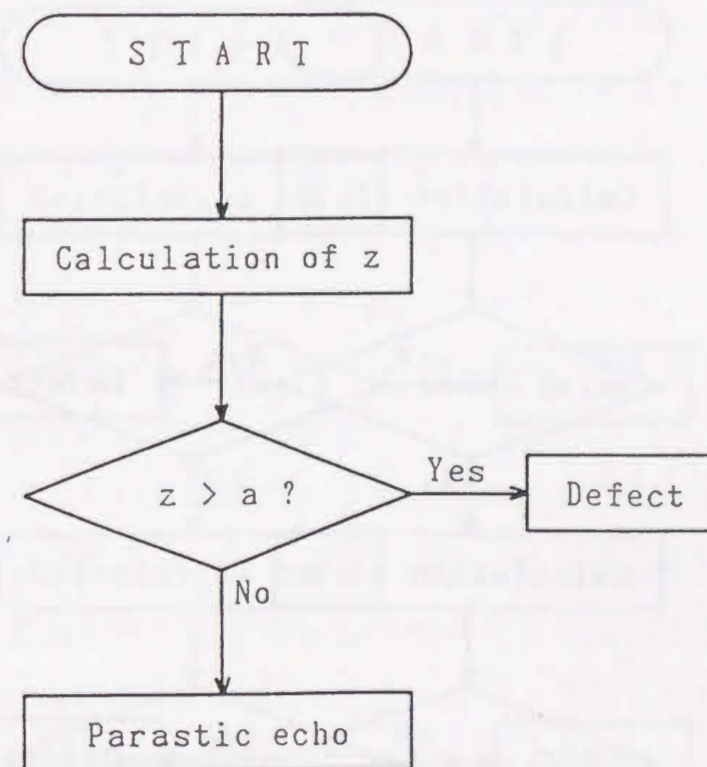


図 5・3 妨害エコー 1 の判定方法



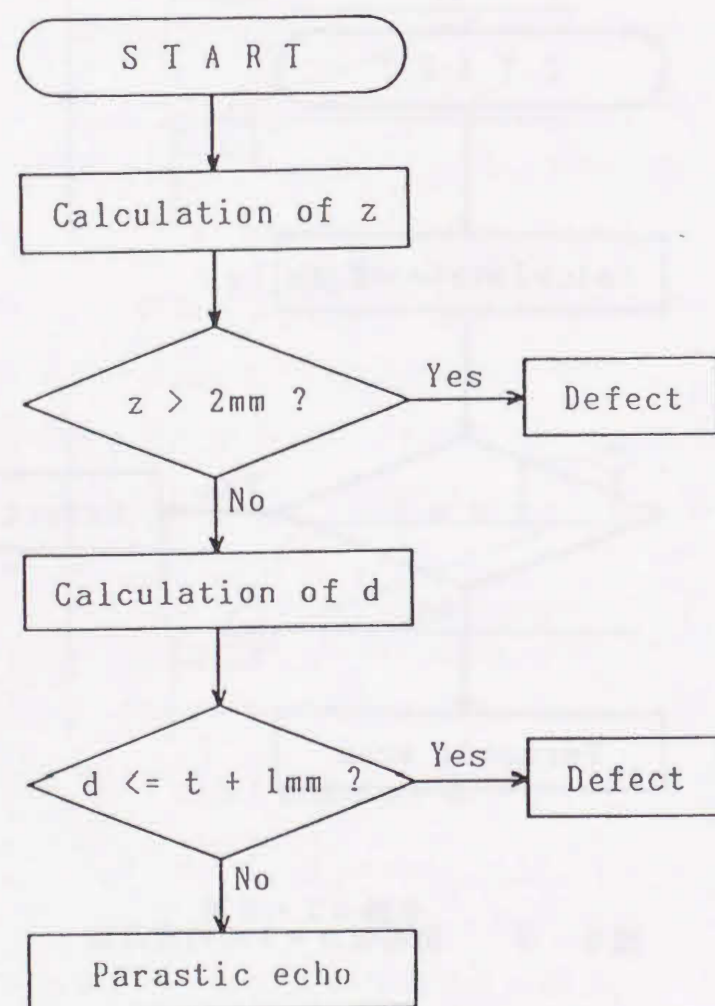
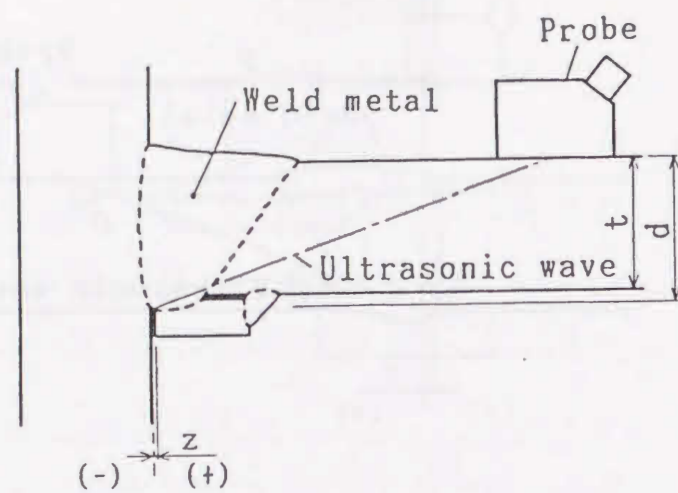


図5・4 妨害エコー7の判定方法

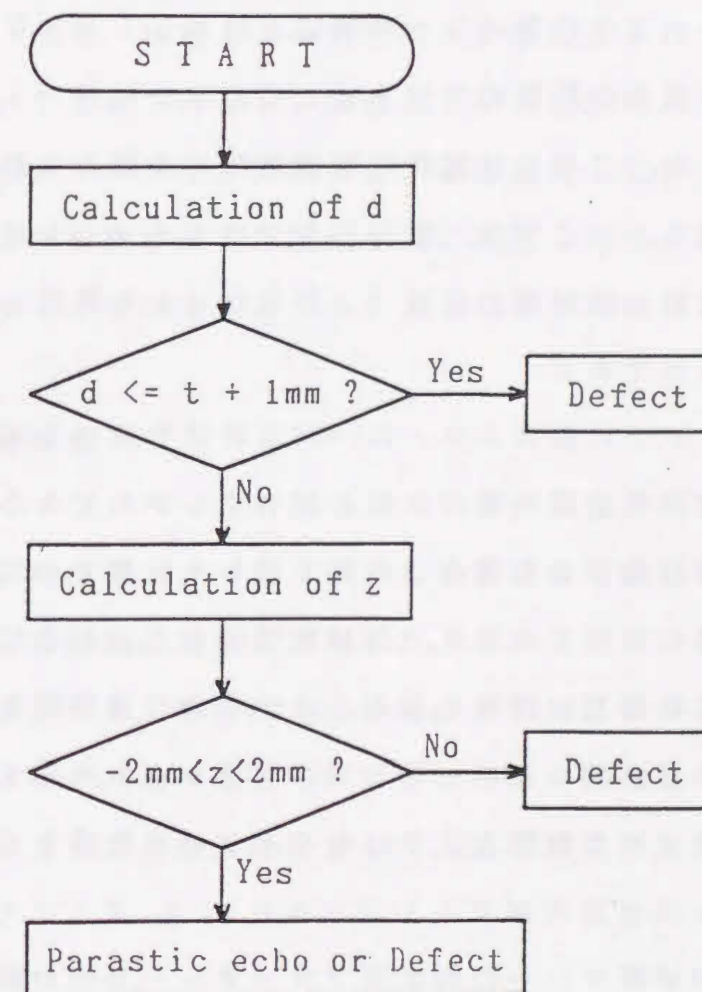
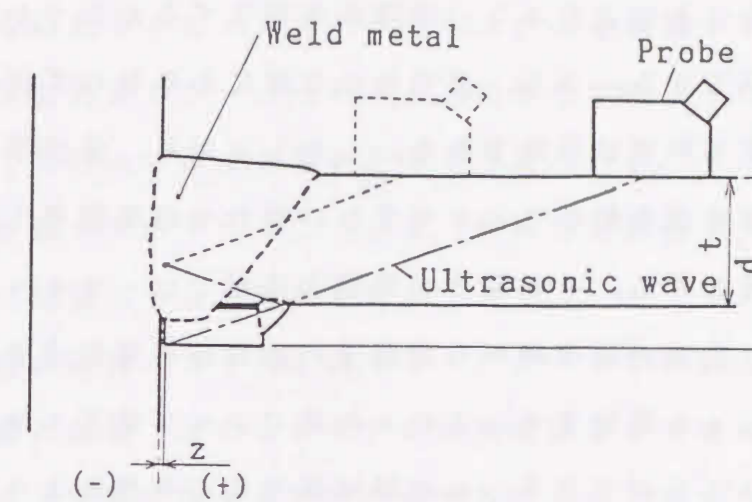


図5・5 妨害エコー8の判定方法



下に、このフローチャートに基づいて説明する。

妨害エコー 1：この妨害エコーの場合、裏当金がない場合には裏波部を指で触ることによりエコーの高さが変化するかどうかによって判別することができる。また、裏当金の有無にかかわらず両側探傷することによっても判別は可能である。しかしながら、溶接部付近の構造の関係から裏波部を触ることができない場合や両側探傷できない場合は判別が困難となる。その場合の判別方法がフローチャートに示した方法である。溶接の基準線から探触子の超音波の発信点までの距離  $y$  を精密測定し  $z$  を算定する。この  $z$  の値によって妨害エコーか欠陥かの判定を行うことができる。 $a$  の値は裏当金の有無によって変わり、裏当金がある場合は 2 mm、無い場合は 0 mm である。

妨害エコー 7：この妨害エコーの判定方法として十分な経験を積んだ検査技術者の最大公約数的方法をまとめたものに N D I 2 0 2 小委員会指針があるが、これには定性的判別項目が多数あり初心者には難しいと考えられる。ここでは、簡易法をフローチャートに示した。妨害エコー 1 と同様に反射源の位置 ( $z$  および  $d$ ) を算出しその値によって判別する方法である。

妨害エコー 8：この妨害エコーは一回反射法を用いる場合に現れるエコーで、通常溶接金属内部の欠陥と誤判定しがちである。直射法によって反射源を探傷できる場合（点線で示した位置での探傷ができる場合）には簡単に判別できるが、溶接の関係から直射法によって探傷できない場合には判別が困難となる。この場合の決定的な判別方法は見いだされていない。したがって、フローチャートに示す判別方法は妨害エコーの決定的な判別方法ではなく、その可能性を示すものである。

以上の 3 つの妨害エコーの判定用フローチャート中の数値は、初心者には装置の調整、 $y$  およびビーム路程の測定に精度上の問題があるため、そのことを考慮して決められている。

5・4・3 知識ベース 前述の 3 つの妨害エコー以外の妨害エコーは熟練者にとっては比較的判別が容易なものであるが、初心者が用いることを前提としたエキスパートシステムを構築する際には比較的判別が容易な妨害エコーについても判別ができるように知識をデータベースとして持っておく必要があると考えられる。そこで、図 5・2 の 8 つの妨害エコーすべてについて、前述した妨害エコーの判別方法などを基に知識ベースを構築した。

一方、前述の妨害エコー判定用フローチャート中の判定を行う際に求める反射源の位置 ( $z$  および  $d$ ) には測定誤差が含まれており、判定部の数値にも実際にはある程度の幅があるものと考えられる。本システムではあいまいなデータや知識を扱うためにファジィ理論を用いた<sup>(18)・(19)</sup>ので、妨害エコー判定用の知識に含まれるこれらのあいまいさについてもメンバシップ関数および確信度を用いて表現する。ここで用いたメンバシップ関数および確信度は上記のフローチャート中の数値および検査技術者の経験を基に決定した後、検査技術者の推定結果に最もよく一致するようにチューニングを行ったものである。

## 5・5 適用結果

前節で示した妨害エコーについての判定用の知識を補充したシステムを実際の超音波探傷試験に適用した。本システムの推論結果の評価の妥当性を確かめるために検査技術者による推定結果と比較したものを表 5・1 に示す。表には参考のために材料の開先形状および板厚を記した。本システムの推論結果として、確信度の大きい欠陥種類と欠陥高さおよび J I S Z 3 0 6 0 による欠陥等級を記した。試験データ数は 2 7 例である。

本システムの推論によって確信度が最大になった欠陥種類（妨害エコーを含む）と検査技術者が推定した欠陥種類が一致したのは 2 7 例



表5・1 本 シ ス テ ム の 適 用 結 果

No.	Profile of Groove	Thick-ness (mm)	Results by Experts (Type of Defect)	Results by This expert system		
				Type of Defect (Certainty factor)	Width of Defect (mm)	Defect class (JIS Z 3060)
1	L	14.0	FB	FB (0.67) GB (0.50) FM (0.45) GM (0.34)	0.12 0.12 0.12 0.12	2
2	V	16.0	FB	FB (0.82) FM (0.74) GB (0.62) GM (0.55)	0.33 0.33 0.33 0.33	3
3	L	16.0	J8	J8 (1.00) FB (0.83) GB (0.62) FM (0.31) I (0.25) GM (0.23)	0.86 0.86 0.86 0.86 0.86	4
4	L	18.0	C	C (1.00) GM (0.25)	2.42 0.13	2
5	L	27.0	J6	J6 (1.00)		3
6	K	28.0	B	B (1.00) FB (0.78) FM (0.60) GB (0.59) GM (0.45) E (0.22)	0.33 0.33 0.33 0.33 0.33 0.33	4
7	L	19.0	FM	FM (0.83) GM (0.78) E (0.30) C (0.22)	0.25 0.25 0.25 5.70	4
8	L	27.0	J3	J3 (1.00)		4
9	L	25.0	A	A (1.00) I (0.30) FB (0.24) FM (0.24)	0.84 0.84 0.84 0.84	4

10	K	28.0	B	B (0.91) FB (0.76) GB (0.71) FM (0.28) H (0.27)	0.31 0.31 0.31 0.31 0.31	4
11	L	30.0	GM	FM (0.93) GM (0.87) C (0.21)	0.36 0.36 11.10	1
12	L	32.0	FM	FM (1.00) GM (0.75) D (0.20)	0.34 0.34 10.06	3
13	L	32.0	J4	J4 (1.00) J3 (0.69)		4
14	L	32.0	J1	J1 (0.77) H (0.30)	0.44	4
15	K	22.0	B	B (0.76) FB (0.56) GB (0.42) H (0.28) FM (0.21)	0.61 0.61 0.61 0.61 0.61	4
16	L	18.0	FM	FM (0.79) GM (0.59) FB (0.48) GB (0.36)	0.28 0.28 0.28 0.28	3
17	X	22.0	FM	FM (0.83) GM (0.78) C (0.24)	0.25 0.25 5.44	2
18	L	16.0	FB	FB (0.80) GB (0.75) FM (0.60) GM (0.56)	0.20 0.20 0.20 0.20	2
19	L	22.0	FM	FM (1.00) GM (0.75) D (0.20)	0.53 0.53 11.20	4
20	V	9.0	J1	J1 (1.00) J2 (0.43) H (0.30) FB (0.25) A (0.19) GB (0.19)	0.24 0.24 0.24 0.24 0.24	4



21	K	75.0	FB	FB (0.52) GB (0.39) B (0.30) H (0.30)	1.23 1.23 1.23 1.23	3
22	K	25.0	GB	FB (0.83) GB (0.78) FM (0.53) GM (0.50) B (0.24)	0.25 0.25 0.25 0.25 0.25	4
23	V	9.0	J2	J2 (0.67)		4
24	K	40.0	C	C (1.00) GB (0.50) B (0.38) GM (0.22) H (0.21) FB (0.20)	6.60 0.27 0.27 0.27 0.27 0.27	4
25	K	18.0	FB	FB (0.67) GB (0.62) FM (0.45) GM (0.42) C (0.22) E (0.20)	0.49 0.49 0.49 0.49 10.47 0.49	4
26	V	21.0	FM	FM (0.81) GM (0.61) FB (0.41) GB (0.31)	0.30 0.30 0.30 0.30	3
27	K	36.0	B	B (1.00) FB (0.83) GB (0.78) FM (0.31) GM (0.29) C (0.22)	0.62 0.62 0.62 0.62 0.62 17.17	4

where A : Lack of joint penetration at root zone  
B : Lack of joint penetration at weld metal  
C : Porosity  
D : Blow hole  
E : Crack at weld metal  
FB : Lack of fusion at weld bond  
FM : Lack of fusion at weld metal  
GB : Slag inclusion at weld bond  
GM : Slag inclusion at weld metal  
H : Crack at heat affected zone  
I : Crack at root zone  
J i : Parastic echo i in Figure 5.2 ( i=1, 2, ... 8 )

中25例であり、精度良く推論できていることが確認された。また、本システムおよび検査技術者共に27例中20例を溶接欠陥、7例を妨害エコーと判定しており、妨害エコーについても本システムは精度良く推論できることが確認された。

以下に、表5・1の中の数例について簡単に説明する。

(1) No. 3は本システムによって確信度が最も大きい欠陥種類として妨害エコー8(J8)が推定されており、検査技術者による推定結果と同じになっているが、そのほかに開先面融合不良(FB)および開先面スラグ巻き込み(GB)の確信度も比較的大きな値となっている。この妨害エコーは前述したように直射法による探傷ができない場合、判別が困難となる。そのため、本システムの推論結果において妨害エコー以外の欠陥種類の確信度が大きく推定されているものと考えられる。また、検査技術者の推定結果には妨害エコーのみが記されているが、実際には探傷試験を行った際、最初融合不良と推定し補修を行ったにもかかわらずエコーに変化がみられなかったことから最終的に妨害エコーと判定しなおしたという経緯がある。そのようなことを考慮すると、本システムの推論結果は妥当なものであると考えられる。

(2) No. 14および20では検査技術者の推定結果の妨害エコー(Ji)に対して本システムでは妨害エコーの確信度が最大になってはいるもののその他の欠陥種類の可能性もあるという結果になっている。このことは、これらのデータが比較的判別が困難な妨害エコー1(J1)に関係しているためと考えられる。

(3) No. 11および22では検査技術者の推定結果がスラグ巻き込み(GM, GB)になっているのに対して本システムでは融合不良(FM, FB)の確信度が最大となっており、両者の結果が一致していない。しかしながら、スラグ巻き込みと融合不良の判別は検査技術者でも難しいとされていることおよび本システムの推論結果において



スラグ巻き込みは2番目に推定されており確信度も大きな値となっていることから、検査技術者とほぼ同程度の推論ができていると判断して差し支えないものと考えられる。

#### 5・6 おわりに

本章では、パソコン用の超音波探傷評価システムを構築すると共に妨害エコーについての判定も行えるように知識の補充を行った。以下に得られた結果を記す。

(1) 効率的な推論を行うために黑板モデルを採用する場合、対象とする問題に用いる知識をいかに体系づけるかが重要となる。特に、推論時間が長くなる恐れのあるパソコン上にシステムを構築する際には十分な検討が必要と考えられる。

(2) 妨害エコー判定用知識を補充し、その知識に含まれているあいまいさをメンバシップ関数および確信度を用いて表しファジィ理論によって扱うことにより、妨害エコーの判定についても精度良く行うことができる超音波探傷試験評価システムを構築することができた。

(3) パソコン用のシステムとしたことで、より実用性の高いシステムとなった。

## 第6章 結 論

本論文では、溶接部の品質を検査する非破壊試験の一つである超音波探傷試験の評価のためのエキスパートシステムの構築に関する研究を行った。その際に、検査技術者の持つ経験的知識や測定データに含まれているあいまいさおよび不確実さについてファジィ理論を用いて扱った。ここで構築したシステムを実際の試験データに適用し、検査技術者による推定結果と比較することによりその有効性について検討した。

第3章では開発した超音波探傷試験評価のためのファジィエキスパートシステムのプロットタイプについて説明した。あいまいさや不確実さを表現するためのメンバシップ関数や確信度を検査技術者から得た経験的知識から作成することにより、推論精度の高いシステムであることが確認された。

第4章では、開発したシステムの推論結果と検査技術者による推定結果との比較から、このシステムに最適なファジィ結合演算子およびメンバシップ関数の形状を決定した。その結果、本システムに最適なファジィ結合演算子として代数積であることが確認された。また、それにより、より推論精度の高いシステムとすることができた。

第5章では、第3章および第4章で開発したシステムをパソコン用システムとして改良すると共に妨害エコーについても推定できるように知識を補充した。これによって、さらに実用的なシステムになったと考えられる。

以上のことから、本研究によって十分実用に耐える超音波探傷試験評価のためのファジィエキスパートシステムが構築できたと考えられる。また、本論文で記したファジィエキスパートシステムの開発手法は、他の分野におけるシステムの開発に十分適用できると思われる。



今後の課題としては、メンバシップ関数の簡単なチューニング方法の開発や推論過程の説明機能およびマンマシンインターフェ이스の充実などが挙げられる。

#### 参 考 文 献

- (1) 日本非破壊検査協会編, 新非破壊検査便覧, (1992), 日刊工業新聞社
- (2) 日本建築学会, 鋼構造建築溶接部の超音波探傷検査規準・同解説, (1989), 日本建築学会
- (3) 寺野・浅井・菅野, 応用ファジィシステム入門, (1989), オーム社
- (4) Buchanan, B. G., G. L. Sutherland and E. A. Feigenbaum, Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry, In B. Meltzer and D. Michie, eds., Machine intelligence, vol. 4, Edinburgh: Edinburgh University Press, (1969), 209-254
- (5) Erman, L. D., F. Hayes-Roth, V. Lesser and D. Reddy, The HEARSAY-II speech-understanding system: Integrating knowledge to resolve uncertainty, Computing Surveys 12, 2, (1980), 213-253
- (6) Davis, R., B. G. Buchanan and E. Shortliffe, Production rules as a representation for a knowledge-based consultation program, Artificial Intelligence, 8, (1977), 15-45
- (7) Weiss, S. M., C. A. Kulikowski and A. Safir, A model-based consultation system for the long-term management of glaucoma, IJGAI, 5, (1977), 826-832
- (8) Martin, W. A. and R. J. Fateman, The MACSYMA system, Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation, (1971), 59-75
- (9) Pople, H. E., Jr., J. D. Myers and R. A. Miller, DIALOG: A model of diagnostic logic for internal medicine, IJCAI, 4, (1975), 848-855



- (10) Pople, H. E., Jr, The formation of composite hypotheses in diagnostic problem solving : An exercise in synthetic reasoning, IJCAI, 5, (1977), 1030-1037
- (11) 村上・林・秋園, 超音波探傷試験方法診断エキスパートシステムの構築の試み, 日本機械学会論文集, 55-511, A(1989), 634-638
- (12) 村上, 超音波探傷による溶接欠陥の検出と評価のためのエキスパート・システム, ジョイテック, 7, (1991), 80-85
- (13) 村上・島田・上田, 超音波探傷試験による溶接欠陥評価のためのエキスパート・システムの構築の試み, 日本機械学会論文集, 59-559, A(1993), 580-587
- (14) 加藤・奥村・糸賀・原田・杉本・道場・井内, 放射線透過写真自動合否判定システムの開発(第1報), 非破壊検査, 41-4, (1992), 186-195
- (15) 加藤・奥村・糸賀・原田・道場・道場・川野, 放射線透過写真自動合否判定システムの開発(第2報), 非破壊検査, 41-4, (1992), 196-206
- (16) 糸賀・道場・黒崎・宮本・西野, 溶接欠陥の種別判定エキスパートシステムの開発, 溶接学会講演概要集, 42, 27-30, (1988)
- (17) 佐々木, エキスパートシステム開発の考え方と非破壊検査への応用, 非破壊検査, 37-10, (1988), 772-779
- (18) 松浦・芳村・川人, 超音波探傷試験評価のためのファジィエキスパートシステム, 日本機械学会論文集, 58-551, C(1992), 2268-2275
- (19) 松浦・芳村・川人, 超音波探傷試験評価のためのファジィエキスパートシステム, 日本機械学会論文集, 58-555, C(1992), 3434-3440
- (20) 松浦・芳村・森本・川人, 超音波探傷試験評価のためのファジィエキスパートシステム, 日本機械学会論文集, 59-566, C(1993),

- 3218-3223
- (21) 橋本, 新版溶接工学, (1980), 297, 工学図書
- (22) 金属の割れ・破壊対策研究会, 金属の割れ・破壊その解析と防止対策, (1980), 349, 経営開発センター出版部
- (23) 榎木・橋・谷口・中山・大道・本村, 次世代型自動超音波探傷試験装置の開発, 火力原子力発電, 42-417, (1991), 784-789
- (24) 古村・鈴木・恩田・杉田, タービンロータ中心孔超音波探傷システムの開発, 火力原子力発電, 43-432, (1992), 1073-1080
- (25) 徳永・青木・馬場・紅林・佐々木・伊藤, 原子炉圧力用超音波探傷装置の開発, 火力原子力発電, 43-432, (1992), 1091-1098
- (26) 原田・二木・古瀬・小路, 軽水炉発電設備運転中診断モニタリング, 火力原子力発電, 42-426, (1991), 595-600
- (27) 宮沢・木村・佐藤, 原子力用自動診断ロボット, 溶接学会誌, 59-3, (1990), 183-189
- (28) 横野, 構造物の余寿命予測に必要な非破壊試験技術, 溶接学会誌, 59-3, (1990), 199-206
- (29) Hayes-Roth, F., Waterman, D. A. and Lenat, D. B., BUILDING EXPERT SYSTEMS, (1983), Addison Wesley  
(邦訳) AIUEO, エキスパートシステム, (1985), 産業図書
- (30) Newell, A., Heuristic programming: Ill-structured problems, A. Aronofsky, ed., Progress in operations research, 3, John Wiley and Sons, 360-414
- (31) 上野, 知識工学入門(改訂2版), (1989), オーム社
- (32) 上野・石塚, 知識の表現と利用, (1987), オーム社
- (33) 村上・興水・長谷川, 画像処理手段のエキスパート知識をガイド情報に持つ会話型溶接欠陥識別システム, 情報処理学会論文誌, 29-2, (1988), 114-125
- (34) 帖佐, 鋼管溶接部の画像表示式超音波探傷装置の開発, 溶接技術,



- 36-9, (1989), 74-78
- (35) 山田, 放射線探傷試験における画像処理, 溶接技術, 37-10, (1989), 82-89
- (36) 村井, 非破壊検査へのエキスパートシステムの応用, 溶接技術, 37-10, (1989), 96-101
- (37) 糸賀・杉本・道場・加藤・杉田・恩田, 画像処理と人工知能を用いた放射線試験合否判定システムの開発, 溶接技術, 38-9, (1990), 102-107
- (38) 福田・相山・大久保・長坂, 非破壊検査要領書作成支援システムの開発, 日本機械学会論文集, 56-521, A(1990), 178-181
- (39) Zadeh, L. A., Fuzzy Sets, Information and Control, 8 (1965), 338-353, New York, Academic Press.
- (40) Zadeh, L. A., Fuzzy Algorithms, Information and Control, 12 (1968), 94-102,
- (41) Zadeh, L. A., Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, IEEE Trans. System, Man and Cybernetics, SMC-3, 1 (1973), 28-44
- (42) Mamdani, E. H. and Assilian, S., An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, Int. J. Man-Machine Studies, 7(1974), 1-13
- (43) Mamdani, E. H., Application of Fuzzy Algorithms for Control of Simple Dynamic Plant, Proc. IEE, 121-12(1974), 1585-1588
- (44) Holmblad, L. P. and Ostergaard, J. J., Control of Cement Kiln by Fuzzy Logic, Fuzzy Information and Decision Processes, (1982), 389-399
- (45) 菅野, ファジィ制御, (1988), 日刊工業新聞社
- (46) 馬野, ファジィエキスパートシステムの現状と展望, 日本ファジィ学会誌, 2-2(1990), 115-124

- (47) 庄野, 非破壊検査エキスパートシステムの構築に関する研究, 徳島大学修士論文, (1989)
- (48) 鬼沢・菅野・西岡・岡田, 歯科診断におけるあいまい推論の利用, 計測自動制御学会論文集, 18-7 (1982), 692-698
- (49) Furuta, H., Fu, K. S. and Yao, J. T. P., Structural engineering applications of expert systems, Computer-Aided Design, 17-9 (1985), 410-419
- (50) 白石・古田・馬野・川上・大谷, 構造信頼性評価のためのファジィエキスパートシステム, 材料, 39-436, (1990), 19-25
- (51) 有田, ファジィ理論を用いた超音波画像による癌診断システムの開発, 日本ファジィ学会誌, 3-3, (1991), 527-539
- (52) 宮本・西村, コンクリート橋の安全性評価と寿命予測, 材料, 40-450 (1991), 263-269
- (53) 古田・馬野・川上・大谷・白石, 構造信頼性評価のためのファジィエキスパートシステムの改良, 材料, 41-466, (1992), 1030-1035
- (54) 岡本・片岡・三好, 欠陥の同定および評価を行うエキスパートシステムの開発, 火力原子力発電, 43-432 (1992), 1059-1066
- (55) 日本非破壊検査協会編, 溶接部の超音波探傷試験, (1979)
- (56) 水本, ファジィ集合とファジィ推論, 第3回ファジィシステムシンポジウム, (1987), 37-48
- (57) Zimmermann, H. J. and Zysno, P., Latent Connectives in Human Decision Making, Fuzzy Sets and Systems, 4, (1980), 37-51
- (58) 前田・村上, ファジィ結合演算による選好表現を用いた多目的問題のファジィ意志決定手法, 計測自動制御学会論文集, 23-5, (1987), 517-524
- (59) 林・内藤・若見, 学習ファジィ論理演算子のあいまい検索への応用, システム制御情報学会第35回研究発表講演会, (1991), 181-182



(60) Siler, W. M., Theory and Practice of Fuzzy Expert System,

(邦訳) 崔訳・廣田監修, ファジィエキスパートシステム 理論と実

践, (1990), 3, 電気書院

## 謝 辞

本論文を執筆するにあたり懇切丁寧なご指導, 貴重なご教示およびご助言を賜り, さらに本論文をご校閲くださいました徳島大学芳村敏夫教授に深甚なる感謝の意を表します。

徳島大学三輪恵教授, 宮本定明教授の先生方にはご多忙のところ本論文の執筆にあたり, 有益なご指摘, ご助言を賜りましたことに深く感謝致します。

研究遂行にあたり, 終始多大なご指導を賜りました徳島大学村上理一教授ならびにシステム構築に際し有益なご意見をご提供くださいました(株)山村鉄工所岡島博代表取締役社長, (株)アルス製作所生産本部岡崎信哉本部長, 四国システム開発(株)システム推進部島田嘉洋部長, 徳島県工業技術センター平尾友二研究員に深く感謝致します。また, 徳島大学大学院工学研究科博士後期課程在学中に, 種々ご指導およびご助力いただきました機械力学研究室の小西克信助教授, 日野順市講師, 森本和憲氏はじめ多数の大学院学生諸氏に対して深く感謝致します。

最後に, 徳島大学大学院工学研究科博士後期課程における研究の機会を与えてくださいました(株)三浦研究所川人明美代表取締役社長ならびにいろいろな面でご協力・ご支援くださいました(株)三浦研究所および三浦工業(株)の皆様には深く感謝致します。



1. 超音波探傷試験の目的と意義  
2. 超音波探傷試験の原理  
3. 超音波探傷試験の装置  
4. 超音波探傷試験の手続き  
5. 超音波探傷試験の結果の解釈  
6. 超音波探傷試験の応用  
7. 超音波探傷試験の注意点  
8. 超音波探傷試験の将来展望

付録 超音波探傷試験評価のための  
ファジィエキスパートシステムのプログラム

```
1. 変数の宣言
2. ファジィ集合の定義
3. 規則の定義
4. 推論の処理
5. デフuzzificationの処理
6. 結果の出力
```



```

**** mulisp86 version ***** 92.11.14 *****
;
; File name : CONTOROL. LSP
; Feature   : システムの始動および推論の制御
;
;*****

(defun start ()
;システムの開始
  (init_data)
  (init_agenda)
  (init_start_frame)
  (start_aux metarule))

(defun init_data ()
  (put '*b_board* 'input_ut* nil)
  (put '*b_board* 'result_w1* nil)
  (put '*b_board* 'result_w2* nil)
  (put '*b_board* 'result_ut1* nil)
  (put '*b_board* 'result_ut2* nil)
  (put '*b_board* 'result_ut3* nil)
  (put '*b_board* 'result_ut* nil)
  (put '*b_board* 'result_ef* nil)
  (put '*b_board* 'result_e* nil)
  (put '*b_board* 'result_ec* nil)
  (put '*result_es* 'frame nil)
  (setq $frame$ nil))

(defun init_agenda ()
  (setq *agenda* nil))

(defun init_start_frame ()
  (setq *start_frame* nil))

(defun start_aux (meta_rule)
;フレームによる制御を開始する
  (cond ((null meta_rule) (done))
        (t (setq *start_frame* (car meta_rule))
            (cycle)
            (start_aux (cdr meta_rule)))))

(defun done ()
  (do ((frame $frame$ (cdr frame)))
      ((null frame) 'done)
      (put (car frame) 'frame nil)))

(defun cycle ()
  (do ()
      ((equal (tick) 'done) t)
      ()))

(defun tick ()
  (cond ((null_agenda) (get_hypothesis)))
  (cond ((null_agenda) 'done)
        (t (setq curframe (get_flag))
            (let ((agenda (car *agenda*)))
              (setq *agenda* (cdr *agenda*))
              (eval agenda))))))

(defun null_agenda ()
  (null (car *agenda*)))

(defun get_hypothesis ()
  (let ((item *start_frame*))
    (init_start_frame)
    (cond ((null item) nil)
          (t (beginframe item)))))

(defun beginframe (frame)
  (let ((method (reverse (fget_i frame 'begin))))
    (cond ((null method)
           (prtsen '(begin スロットが見つかりません)))
          (t (add_method frame method)))))

(defun add_method (frame method)
  (cond ((null method) t)
        (t (set_flag frame (car method))
            (push_method (list (car method)))
            (add_method frame (cdr method)))))

(defun set_flag (frame method)
  (put 'flag method frame))

(defun push_method (method)
  (push method *agenda*))

(defun get_flag ()
  (get 'flag (caar *agenda*)))

(rds b:ndi_m.kdb)

```

```

**** mulisp86 version ***** 92.11.14 *****
;
; File name : METHOD. LSP
; Feature   : 各フレームに対するMETHOD
;
;*****

;---[ method for all_frames ]-----
;subclass スロットに格納されたフレーム名のフレームに制御をわたす

(defun subclass ()
  (let ((frames (reverse (fget_v_d curframe 'subclasses)))) ;フレーム名
    (subclass_aux frames)))

(defun subclass_aux (frame)
  (cond ((null frame) t)
        (t (beginframe (car frame))
            (subclass_aux (cdr frame)))))

;---[ method for get_data frame ]-----
;溶接条件および超音波探傷に関するデータを入力する

(defun get_d ()
;フレームから質問文を入力し、order リストに基づいてデータを入力する
  (let ((result1 (car (fget_c 'result1)))
        (result2 (car (fget_c 'result2)))
        (identify (fget_c 'identify))
        (order_list (fget_c 'order_list))
        (ndata_list (fget_c 'get_ndata_list))
        (vdata_list (fget_c 'get_vdata_list))
        (judge_list (fget_c 'judg_list))
        (data2_list (fget_c 'result2_data)))
    (clear-screen)
    (set-cursor)
    (put '*b_board* result1 nil)
    (put '*b_board* result2 nil)
    (prtsen identify)
    (cond ((null order_list)
           (print '(order リストが見つかりません)))
          (t (get_d_aux result1 result2 order_list ndata_list
                        vdata_list judge_list data2_list)
              (cond ((y-or-n-p "溶接条件の入力データを修正しますか")
                     (get_d))
                    (t) )))))

(defun get_d2 ()
;フレームから質問文を入力し、order リストに基づいてデータを入力する
  (let ((result1 (car (fget_c 'result1)))
        (result2 (car (fget_c 'result2)))
        (identify (fget_c 'identify))
        (order_list (fget_c 'order_list))
        (ndata_list (fget_c 'get_ndata_list))
        (vdata_list (fget_c 'get_vdata_list))
        (judge_list (fget_c 'judg_list))
        (data2_list (fget_c 'result2_data)))
    (clear-screen)
    (set-cursor)
    (put '*b_board* result1 nil)
    (put '*b_board* result2 nil)
    (prtsen identify)
    (cond ((null order_list)
           (print '(order リストが見つかりません)))
          (t (get_d_aux result1 result2 order_list ndata_list
                        vdata_list judge_list data2_list)
              (cond ((y-or-n-p "超音波探傷の入力データを修正しますか")
                     (get_d2))
                    (t (clear-screen)
                        (set-cursor 12 32)
                        (prtsen '(推論中です)) )))))))

(defun get_d_aux (result1 result2 order ndata vdata judge data2)
;order リストの順序に基づくデータの入力
  (cond ((null order) t)
        (t (get_nv_data result1 result2 (car order) ndata vdata data2)
            (judge_data result1 result2 (car order) ndata vdata judge data2)
            (get_d_aux result1 result2 (cdr order) ndata vdata judge data2))))

(defun judge_data (result1 result2 name ndata vdata judge data2)
;入力データに基づき judge リストを利用して質問事項を選択する
  (cond ((null judge) t)
        ((equal name (caar judge))
         (let ((slot_name (search_slot_name result1 result2 name
                                             (cdr (car judge)) data2)))
           (judge_data_aux result1 result2 (fget_c slot_name)

```



```

                                ndata vdata (cdr judge) data2)))
    (t (judge_data result1 result2 name ndata vdata (cdr judge) data2)))

(defun judge_data_aux (result1 result2 name_list ndata vdata judge data2)
  ;質問事項が複数であるとき
  ;処理が終了するとtを返す
  (cond ((null name_list) t)
        (t (get_nv_data result1 result2 (car name_list) ndata vdata data2)
            (judge_data result1 result2 (car name_list) ndata vdata judge data2)
            (judge_data_aux result1 result2 (cdr name_list) ndata vdata judge
                             data2) )))

(defun get_nv_data (result1 result2 name ndata vdata data2)
  ;質問項目の選択 (数値データまたは言語データ)
  (cond ((search_name name ndata)
        (get_ndata result1 result2 name data2))
        ((search_name name vdata)
        (get_vdata result1 result2 name data2))
        (t (print '(スロットが見つかりません)))))

(defun get_ndata (result1 result2 name data2)
  ;言語データの入力
  (let ((question (car (fget_c name)))
        (answer (cdr (fget_c name)))
        (message (cons question answer))
        (choice_n result1 result2 name question answer (read) data2)))

    (defun search_slot_name (result1 result2 name a_list data2)
      ;* b _ b o a r d * から入力値を読みだし値を照合する関数にデータを送る
      (let ((value (search_value name (cond ((search_name name data2) result2)
                                             (t result1))))
            (search_slot_name_aux value a_list)))

    (defun search_slot_name_aux (value a_list)
      ;入力された値をj u d g eリストと照合する
      (cond ((null a_list) nil)
            ((equal (caar a_list) value)
             (cadr (car a_list)))
            (t (search_slot_name_aux value (cdr a_list)))))

    (defun search_name (name a_list)
      ;リスト中のa t o mの検索
      (cond ((null a_list) nil)
            ((equal (car a_list) name) t)
            (t (search_name name (cdr a_list)))))

    (defun search_value (name data)
      ;データ中の値を検索
      (do ((datal (get '*b_board* data) (cdr datal)))
          ((or (null datal)(equal (caadar datal) name)) (caddar (cdar datal)))
          ( )))

    (defun b_push (rule data name value weight)
      ;データを確信度の大きい順に* b _ b o a r d *に格納する
      (cond ((null weight)(setq weight 1.0)))
      (cond ((search_name name (fget 'get_data_unit 'name 'value))
            (put '*b_board* data
                 (remember (get '*b_board* data)
                           rule
                           (list name 'は value (car (fget name 'unit 'value)))
                           weight) )))
            (t (put '*b_board* data
                    (remember (get '*b_board* data)
                              rule
                              (list name 'は value)
                              weight) )))

    (defun get_vdata (result1 result2 name data2)
      ;数値データを入力し* b _ b o a r d *に書き込む
      (let ((question (list (fget_c name)))
            (cond ((or (null name) (null question)) nil)
                  (t (message question)
                     (let ((response (read)))
                       (cond ((not (numberp response))
                             (prtsen '(もう一度正しい値を入力して下さい。))
                             (get_vdata result1 result2 name data2))
                           (t (cond ((search_name name data2)
                                     (b_push nil result2 name response 1.0))
                                     (t (b_push nil result1 name response 1.0))
                                     ))))))))

    (defun message (messages)
      ;質問文の表示
      (print_char messages)
      (prin1 '--))

    (defun choice_n (result1 result2 name question answer_list answer data2)
      ;質問項目の選択
      (let ((number '1))
        (let ((flag_n (choice_n1 result1 result2 name answer_list number

```

```

                                answer data2)))
        (cond ((null flag_n)
              (prtsen '(もう一度正しい値を入力して下さい))
              (get_ndata result1 result2 name data2))
              (t t))))

(defun choice_n1 (result1 result2 name messages number answer data2)
  ;言語データを* b _ b o a r d *に書き込む
  (cond ((null messages) nil)
        ((equal answer number)
         (cond ((search_name name data2)
               (b_push nil result2 name (cadar messages) 1.0))
               (t (b_push nil result1 name (cadar messages) 1.0))))
        (t (choice_n1 result1 result2 name (cdr messages) (+ 1 number)
                       answer data2))))

```

---[ method for t\_rules ]-----

;前向き推論を実行する

```

(defun t_rules ()
  (let ((rules_c (fget_v_d curframe 'rules))
        (forward_name (fget_v_d curframe 'forward))
        (input_name (fget_v_d curframe 'input))
        (result_name (car (fget_v_d curframe 'result)))
        (whylog_name (car (fget_v_d curframe 'whylog))))
    (setq forward forward_name)
    (setq input input_name)
    (setq result result_name)
    (setq rules_used whylog_name)
    (forward_chain rules_c)))

```

```

(defun t_rules2 ()
  (let ((rules_c (fget_v_d curframe 'rules))
        (forward_name (fget_v_d curframe 'forward))
        (input_name (fget_v_d curframe 'input))
        (result_name (car (fget_v_d curframe 'result)))
        (whylog_name (car (fget_v_d curframe 'whylog))))
    (setq forward forward_name)
    (setq input input_name)
    (setq result result_name)
    (setq rules_used whylog_name)
    (forward_chain rules_c)))

```

---[ method for choice\_groove ]-----

;開先形状に対応するデータ変換フレームを選択する

```

(defun choice_g ()
  (let ((groove (search_value '開先形状 '*input_w1*))
        (let ((frame (car (fget_v_d curframe groove))))
          (beginframe frame))))

```

---[ method for data\_transform ]-----

;超音波探傷試験のデータより欠陥の位置を決定  
;点数の計算  
;E高さと欠陥指示長さと溶接材の板厚を欠陥の等級判定用のデータに変換

```

(defun data_t ()
  (let ((groove (car (fget_v_d curframe 'groove_type))))
    (cond ((equal groove 'single_v)
          (data_t_sv))
          ((equal groove 'double_v)
          (data_t_dv))
          ((equal groove 'single_b)
          (data_t_sb))
          ((equal groove 'double_b)
          (data_t_db)))
    (data_t_l)
    (data_t_t_l)))

```

```

(defun data_t_sv ()
  ;V形開先形状と超音波探傷のデータより欠陥の位置を決定する
  ;データの入力
  (let ((r_angle (search_value 屈折角 '*input_ut2*))
        (face (search_value 探傷面 '*input_ut2*))
        (w (search_value B路程 '*input_ut2*))
        (y (search_value Y '*input_ut2*))
        (t (search_value 板厚 '*input_w2*))
        (gap (search_value R間隔 '*input_w2*))
        (depth (search_value 開先深 '*input_w2*))
        (angle (search_value 開先角 '*input_w2*))
        (haturi (search_value 裏はつり '*input_w1*))
        (let ((z (calculation_z y w t r_angle))) ;Zの決定

```



```

(d (calculation_d w t r_angle))) ;Dの決定
(b_push nil '*input_ut* 'Z (r10 z) 1.0)
(b_push nil '*input_ut* 'D (r10 d) 1.0)
(let ((dd (cond ((equal face '表面)
                  (v_parastic y w t r_angle (/ gap 2) haturi)
                  d)
              (t (- t d)))))
  (cond ((<= depth dd)
        (root_haz_sv
         (abs z) dd t (/ gap 2) depth))
        ;欠陥位置の決定
        ; (ルート部, H A Z部)
        (t (deposit_groove_haz_sv
              (abs z) dd t (/ gap 2) depth (/ angle 2))))))
        ;欠陥位置の決定
        ; (溶接金属部, 開先部, H A Z部)

(defun data_t_dv ()
;X形開先形状と超音波探傷のデータより欠陥の位置を決定する
  (let ((r_angle (search_value 屈折角 *input_ut2*)) ;データの入力
        (face (search_value 探傷面 *input_ut2*))
        (w (search_value B路程 *input_ut2*))
        (y (search_value Y *input_ut2*))
        (t (search_value 板厚 *input_w2*))
        (gap (search_value R間隔 *input_w2*))
        (u_depth (search_value 表開先深 *input_w2*))
        (l_depth (search_value 裏開先深 *input_w2*))
        (u_angle (search_value 表開先角 *input_w2*))
        (l_angle (search_value 裏開先角 *input_w2*)))
    (let ((z (calculation_z y w t r_angle))) ;Zの決定
          (d (calculation_d w t r_angle))) ;Dの決定
          (b_push nil '*input_ut* 'Z (r10 z) 1.0)
          (b_push nil '*input_ut* 'D (r10 d) 1.0)
          (let ((dd (cond ((equal face '表面) d)
                          (t (- t d)))))
            (cond ((and (<= u_depth dd) (>= (- t l_depth) dd))
                  (deposit_haz_dv
                   (abs z) dd t (/ gap 2) u_depth l_depth))
                  ;欠陥位置の決定
                  ; (ルート部, H A Z部)
                  ((> u_depth dd)
                   (deposit_groove_haz_dv
                    (abs z) dd t (/ gap 2) u_depth (/ u_angle 2)))
                   ;欠陥位置の決定
                   ; (溶接金属部, 上開先部, H A Z部)
                   (t (deposit_groove_haz_dv
                        (abs z) (- t dd) t (/ gap 2) l_depth
                        (/ l_angle 2))))))
                  ;欠陥位置の決定
                  ; (溶接金属部, 下開先部, H A Z部)

(defun data_t_sb ()
;レ形開先形状と超音波探傷のデータより欠陥の位置を決定する
  (let ((r_angle (search_value 屈折角 *input_ut2*)) ;データの入力
        (face (search_value 探傷面 *input_ut2*))
        (side (search_value 探傷子 *input_ut2*))
        (w (search_value B路程 *input_ut2*))
        (y (search_value Y *input_ut2*))
        (t (search_value 板厚 *input_w2*))
        (gap (search_value R間隔 *input_w2*))
        (depth (search_value 開先深 *input_w2*))
        (angle (search_value 開先角 *input_w2*))
        (haturi (search_value 裏はつり *input_w1*)))
    (let ((z (calculation_z y w t r_angle))) ;Zの決定
          (d (calculation_d w t r_angle))) ;Dの決定
          (b_push nil '*input_ut* 'Z (r10 z) 1.0)
          (b_push nil '*input_ut* 'D (r10 d) 1.0)
          (cond ((equal face '表面)
                  (cond ((equal side '開先を施した材の上)
                        (l_parastic1 y w t r_angle gap haturi))
                      (t (l_parastic2 y w t r_angle gap haturi))))
                  (let ((dd (cond ((equal face '表面) d)
                                  (t (- t d)))))
                    (zz (cond ((equal side '開先を施した材の上) z)
                            (t (- z)))))
                    (cond ((<= depth dd)
                          (root_haz_sb zz dd t gap depth))
                          ;欠陥位置の決定
                          ; (ルート部, H A Z部)
                          (t (deposit_groove_haz_sb zz dd t gap depth
                                angle))))))
                          ;欠陥位置の決定
                          ; (溶接金属部, 開先部, H A Z部)

(defun data_t_db ()
;K形開先形状欠陥の位置を決定する
  (let ((r_angle (search_value 屈折角 *input_ut2*)) ;データの入力

```

```

(face (search_value 探傷面 *input_ut2*))
(side (search_value 探傷子 *input_ut2*))
(w (search_value B路程 *input_ut2*))
(y (search_value Y *input_ut2*))
(t (search_value 板厚 *input_w2*))
(gap (search_value R間隔 *input_w2*))
(u_depth (search_value 表開先深 *input_w2*))
(l_depth (search_value 裏開先深 *input_w2*))
(u_angle (search_value 表開先角 *input_w2*))
(l_angle (search_value 裏開先角 *input_w2*))
(let ((z (calculation_z y w t r_angle))) ;Zの決定
      (d (calculation_d w t r_angle))) ;Dの決定
      (b_push nil '*input_ut* 'Z (r10 z) 1.0)
      (b_push nil '*input_ut* 'D (r10 d) 1.0)
      (let ((dd (cond ((equal face '表面) d)
                      (t (- t d)))))
        (zz (cond ((equal side '開先を施した材の上) z)
                  (t (- z)))))
        (cond ((and (>= dd u_depth) (<= dd (- t l_depth)))
              (deposit_haz_db zz dd t gap u_depth))
              ;欠陥位置の決定
              ; (溶接金属部, H A Z部)
              ((> u_depth dd)
               (deposit_groove_haz_db zz dd t gap u_depth
                                       u_angle))
               ;欠陥位置の決定
               ; (溶接金属部, 上開先部, H A Z部)
               (t (deposit_groove_haz_db zz (- t dd) t gap l_depth
                                       l_angle))))))
               ;欠陥位置の決定
               ; (溶接金属部, 下開先部, H A Z部)

(defun calculation_z (y w t angle)
;Zの計算
  (- y (* w (sin (radian angle)))) )

(defun calculation_d (w t angle)
;Dの計算
  (let ((d0 (* w (cos (radian angle)))))
    (cond ((< d0 t)
           (t (- (* 2 t) d0)))
          ;直射の時
          ;1回反射の時

(defun v_parastic (y w t angle gap haturi)
;V形開先妨害エコー判定用データ
  (let ((cos0 (cos (radian angle)))
        (d0 (* w (cos (radian angle))))
        (y0 (- y (* t (tan (radian angle)))))
        (z0 (- y (* w (sin (radian angle)))))
        (gap2 (cond ((equal haturi 'する) 5)
                     (t 0))))
    (let ((ref1Weight (max (min (+ 1 (* (- d0 t 0.5) (/ 1 2.0)))
                              1)
                          0))
          (ref2Weight (max (min (+ 1 (* (- d0 t 0.5) (/ 1 1.0)))
                              (- 1 (* (- d0 t 2.0) (/ 1 1.0)))
                              (- 1 (* (+ z0 gap 0.5) (/ 1 1.0)))
                              (+ 1 (* (+ z0 gap 3.5 gap2) (/ 1 2.0)))
                              1)
                          0))
          (ref3Weight (max (min (+ 1 (* (- d0 t (* t cos0) -1.0) (/ 1 2.0)))
                              (- 1 (* (- d0 t (* t cos0) 2.0) (/ 1 4.0)))
                              1)
                          0))
          (ref4Weight (max (min (- 1 (* (- y0 2.0) (/ 1 3.0)))
                              (+ 1 (* (+ y0 2.0 gap2) (/ 1 3.0)))
                              1)
                          0))
          (ref5Weight (max (min (+ 1 (* (- d0 t 3.0) (/ 1 2.0)))
                              (- 1 (* (- d0 t 15.0) (/ 1 5.0)))
                              (- 1 (* (+ z0 (* 4 gap) (/ 1 4.0)))
                              (+ 1 (* (+ z0 (* 8 gap) (/ 1 4.0)))
                              1)
                          0))
          (ref6Weight (max (min (+ 1 (* (- d0 t 1.0) (/ 1 1.0)))
                              (- 1 (* (- d0 t 5.0) (/ 1 3.0)))
                              (- 1 (* (+ z0 (* 6 gap) (/ 1 4.0)))
                              (+ 1 (* (+ z0 (* 10 gap) (/ 1 4.0)))
                              1)
                          0))
          (ref7Weight (max (min (- 1 (* (- y0 gap 3.5) (/ 1 2.0)))
                              (+ 1 (* (+ y0 gap 3.5) (/ 1 2.0)))
                              1)
                          0))
          (ref8Weight (max (min (- 1 (* (- y0 (* 10 gap) (/ 1 2.0)))
                              (+ 1 (* (- y0 (* 2 gap) (/ 1 2.0)))
                              1)
                          0))
          (ref9Weight (max (min (+ 1 (* (- d0 t 0.5) (/ 1 1.0)))

```



```

(- 1 (* (- d0 t 2.0 ) (/ 1 1.0)))
(+ 1 (* (- z0 (* 2 gap)) (/ 1 2.0)))
(- 1 (* (- z0 (* 8 gap)) (/ 1 4.0)))
1)
0)))
(b_push nil '*result_ut3* '路程長さ '1 回反射である ref1Weight)
(b_push nil '*result_ut1* '欠陥位置 '裏波付近である ref2Weight)
(b_push nil '*result_ut1* '欠陥位置 '表面付近である ref3Weight)
(b_push nil '*result_ut3* '反射点 '裏波付近である ref4Weight)
(b_push nil '*result_ut1* '欠陥位置 '裏当金付近である ref5Weight)
(b_push nil '*result_ut1* '欠陥位置 '裏当金溶接裏波付近である
ref6Weight)
(b_push nil '*result_ut3* '反射点 'ルート部である ref7Weight)
(b_push nil '*result_ut3* '反射点 '裏当金溶接付近である
ref8Weight)
(b_push nil '*result_ut1* '欠陥位置 '裏当金溶接付近である
ref9Weight)))))

(defun l_parastic1 (y w t angle gap haturi)
;レ形開先妨害エコー判定用データ1
(let ((cos0 (cos (radian angle))))
(d0 (* w (cos (radian angle))))
(y0 (- y (* t (tan (radian angle)))))
(z0 (- y (* w (sin (radian angle)))))
(gap2 (cond ((equal haturi 'する) 5)
(t 0))))
(let ((ref1Weight (max (min (+ 1 (* (- d0 t 0.5) (/ 1 2.0)))
1)
0))
(ref2Weight (max (min (+ 1 (* (- d0 t 0.5) (/ 1 1.0)))
(- 1 (* (- d0 t 2.0) (/ 1 1.0)))
(- 1 (* (+ z0 0.5) (/ 1 1.0)))
(+ 1 (* (+ z0 3.5 gap2) (/ 1 2.0)))
1)
0))
(ref3Weight (max (min (+ 1 (* (- d0 t (* t cos0) -1.0) (/ 1 2.0)))
(- 1 (* (- d0 t (* t cos0) 2.0) (/ 1 4.0)))
1)
0))
(ref4Weight (max (min (- 1 (* (- y0 (/ gap 2) 2.0) (/ 1 3.0)))
(+ 1 (* (- y0 (/ gap 2) -1.0 gap2) (/ 1 3.0)))
1)
0))
(ref5Weight (max (min (+ 1 (* (- d0 t 3.0) (/ 1 2.0)))
(- 1 (* (- d0 t 15.0) (/ 1 5.0)))
(- 1 (* (+ z0 (* 1.5 gap)) (/ 1 4.0)))
(+ 1 (* (+ z0 (* 3.5 gap)) (/ 1 4.0)))
1)
0))
(ref6Weight (max (min (+ 1 (* (- d0 t 1.0) (/ 1 1.0)))
(- 1 (* (- d0 t 5.0) (/ 1 3.0)))
(- 1 (* (+ z0 (* 3 gap)) (/ 1 4.0)))
(+ 1 (* (+ z0 (* 5 gap)) (/ 1 4.0)))
1)
0))
(ref7Weight (max (min (- 1 (* (- y0 gap 3.5) (/ 1 2.0)))
(+ 1 (* (+ y0 3.5) (/ 1 2.0)))
1)
0))
(ref8Weight (max (min (- 1 (* (- y0 (* 5.5 gap)) (/ 1 2.0)))
(+ 1 (* (- y0 (* 1.5 gap)) (/ 1 2.0)))
1)
0))
(ref9Weight (max (min (+ 1 (* (- d0 t 0.5) (/ 1 1.0)))
(- 1 (* (- d0 t 2.0) (/ 1 1.0)))
(+ 1 (* (- z0 (* 1.5 gap)) (/ 1 2.0)))
(- 1 (* (- z0 (* 5.5 gap)) (/ 1 4.0)))
1)
0))
(ref10Weight (max (min (- 1 (* (+ z0 -1.5) (/ 1 1.0)))
(+ 1 (* (+ z0 1.5) (/ 1 1.0)))
(+ 1 (* (- d0 t 0.5) (/ 1 1.0)))
(- 1 (* (- d0 t 2.0) (/ 1 1.0)))
1)
0))
(ref11Weight (max (min (- 1 (* (+ z0 -1.5) (/ 1 1.0)))
(+ 1 (* (+ z0 1.5) (/ 1 1.0)))
1)
0)))
(b_push nil '*result_ut3* '路程長さ '1 回反射である ref1Weight)
(b_push nil '*result_ut1* '欠陥位置 '裏波付近である ref2Weight)
(b_push nil '*result_ut1* '欠陥位置 '表面付近である ref3Weight)
(b_push nil '*result_ut3* '反射点 '裏波付近である ref4Weight)
(b_push nil '*result_ut1* '欠陥位置 '裏当金付近である
ref5Weight)
(b_push nil '*result_ut1* '欠陥位置 '裏当金溶接裏波付近である
ref6Weight))

```

```

(b_push nil '*result_ut3* '反射点 'ルート部である ref7Weight)
(b_push nil '*result_ut3* '反射点 '裏当金溶接付近である
ref8Weight)
(b_push nil '*result_ut1* '欠陥位置 '裏当金溶接付近である
ref9Weight)
(b_push nil '*result_ut1* '欠陥位置 '開先裏波付近である
ref10Weight)
(b_push nil '*result_ut1* '欠陥位置 '開先付近である ref11Weight)))))

(defun l_parastic2 (y w t angle gap haturi)
;レ形開先妨害エコー判定用データ2
(let ((cos0 (cos (radian angle))))
(d0 (* w (cos (radian angle))))
(y0 (- y (* t (tan (radian angle)))))
(z0 (- y (* w (sin (radian angle)))))
(gap2 (cond ((equal haturi 'する) 5)
(t 0))))
(let ((ref1Weight (max (min (+ 1 (* (- d0 t 0.5) (/ 1 2.0)))
1)
0))
(ref2Weight (max (min (+ 1 (* (- d0 t 0.5) (/ 1 1.0)))
(- 1 (* (- d0 t 2.0) (/ 1 1.0)))
(- 1 (* (+ z0 gap 0.5) (/ 1 1.0)))
(+ 1 (* (+ z0 gap 3.5 gap2) (/ 1 2.0)))
1)
0))
(ref3Weight (max (min (+ 1 (* (- d0 t (* t cos0) -1.0) (/ 1 2.0)))
(- 1 (* (- d0 t (* t cos0) 2.0) (/ 1 4.0)))
1)
0))
(ref4Weight (max (min (- 1 (* (+ y0 -2.0 (/ gap 2)) (/ 1 3.0)))
(+ 1 (* (+ y0 2.0 (/ gap 2) gap2) (/ 1 3.0)))
1)
0))
(ref5Weight (max (min (+ 1 (* (- d0 t 3.0) (/ 1 2.0)))
(- 1 (* (- d0 t 15.0) (/ 1 5.0)))
(- 1 (* (+ z0 (* 2.5 gap)) (/ 1 4.0)))
(+ 1 (* (+ z0 (* 4.5 gap)) (/ 1 4.0)))
1)
0))
(ref6Weight (max (min (+ 1 (* (- d0 t 1.0) (/ 1 1.0)))
(- 1 (* (- d0 t 5.0) (/ 1 3.0)))
(- 1 (* (+ z0 (* 4 gap)) (/ 1 4.0)))
(+ 1 (* (+ z0 (* 6 gap)) (/ 1 4.0)))
1)
0))
(ref7Weight (max (min (- 1 (* (- y0 3.5) (/ 1 2.0)))
(+ 1 (* (+ y0 gap 3.5) (/ 1 2.0)))
1)
0))
(ref8Weight (max (min (- 1 (* (- y0 (* 4.5 gap)) (/ 1 2.0)))
(+ 1 (* (- y0 (* 0.5 gap)) (/ 1 2.0)))
1)
0))
(ref9Weight (max (min (+ 1 (* (- d0 t 0.5) (/ 1 1.0)))
(- 1 (* (- d0 t 2.0) (/ 1 1.0)))
(+ 1 (* (- z0 (* 0.5 gap)) (/ 1 2.0)))
(- 1 (* (- z0 (* 4.5 gap)) (/ 1 4.0)))
1)
0)))
(b_push nil '*result_ut3* '路程長さ '1 回反射である ref1Weight)
(b_push nil '*result_ut1* '欠陥位置 '裏波付近である ref2Weight)
(b_push nil '*result_ut1* '欠陥位置 '表面付近である ref3Weight)
(b_push nil '*result_ut3* '反射点 '裏波付近である ref4Weight)
(b_push nil '*result_ut1* '欠陥位置 '裏当金付近である ref5Weight)
(b_push nil '*result_ut1* '欠陥位置 '裏当金溶接裏波付近である
ref6Weight)
(b_push nil '*result_ut3* '反射点 'ルート部である ref7Weight)
(b_push nil '*result_ut3* '反射点 '裏当金溶接付近である
ref8Weight)
(b_push nil '*result_ut1* '欠陥位置 '裏当金溶接付近である
ref9Weight)))))

(defun root_haz_sv (z d t gap depth)
;V形開先の欠陥位置の決定 (ルート部, HAZ部)
(let ((depositWeight (max (min (- 1 (* (- z gap -2) (/ 1 5.5)))
(- 1 (* (- d depth -1) (/ 1 2.0)))
1)
0))
(grooveWeight (max (min (+ 1 (* (- z gap -0.4) (/ 1 0.9)))
(- 1 (* (- z gap 1.5) (/ 1 2.0)))
(- 1 (* (- d depth -1) (/ 1 2.0)))
1)
0))
(rootWeight (max (min (- 1 (* (- z gap 1.5) (/ 1 2.0)))
1)
0)))

```







```

      gap 2.0)/(1 3.0)))
      (- 1 (* (- z (* (- depth d)(tan (radian angle)))
      gap 4.0)/(1 2.0)))
      (min (- 1 (* (+ z 2.0)/(1 3.0)))
      (+ 1 (* (+ z 4.0)/(1 2.0)))
      0)
      1))
      (baseWeight
      (min (max
      (+ 1 (* (- z (* (- depth d)(tan (radian angle)))
      gap 4.0)/(1 3.0)))
      (- 1 (* (+ z 4.0)/(1 3.0)))
      0)
      1)))
      (b_push nil '*result_ut1* 欠陥位置 溶接金属部である depositWeight)
      (b_push nil '*result_ut1* 欠陥位置 開先部である grooveWeight)
      (b_push nil '*result_ut1* 欠陥位置 ルート部である rootWeight)
      (b_push nil '*result_ut1* 欠陥位置 HAZ部である hazWeight)
      (b_push nil '*result_ut1* 欠陥位置 母材部である baseWeight)))

(defun deposit_haz_db (z d t gap depth)
;K形開先の欠陥位置の決定 (溶接金属部, HAZ部)
  (let ((depositWeight (max (min (+ 1 (* (+ z -2.0)/(1 5.5)))
      (- 1 (* (- z gap -2.0)/(1 5.5)))
      1)
      0))
      (grooveWeight (min (max (+ 1 (* (- z gap -0.4)/(1 0.9)))
      (- 1 (* (- z gap 1.5)/(1 2.0)))
      (min (- 1 (* (+ z -0.4)/(1 0.9)))
      (+ 1 (* (+ z 1.5)/(1 2.0)))
      0)
      1))
      (rootWeight (max (min (- 1 (* (- z gap 1.6)/(1 0.9)))
      (+ 1 (* (+ z 1.6)/(1 0.9)))
      1)
      0))
      (hazWeight (min (max (min (+ 1 (* (- z gap 2.0)/(1 3.0)))
      (- 1 (* (- z gap 4.0)/(1 2.0)))
      (min (- 1 (* (+ z 2.0)/(1 3.0)))
      (+ 1 (* (+ z 4.0)/(1 2.0)))
      0)
      1))
      (baseWeight (min (max (+ 1 (* (- z gap 4.0)/(1 3.0)))
      (- 1 (* (+ z 4.0)/(1 3.0)))
      0)
      1)))
      (b_push nil '*result_ut1* 欠陥位置 溶接金属部である depositWeight)
      (b_push nil '*result_ut1* 欠陥位置 開先部である grooveWeight)
      (b_push nil '*result_ut1* 欠陥位置 溶接金属ルート部である rootWeight)
      (b_push nil '*result_ut1* 欠陥位置 HAZ部である hazWeight)
      (b_push nil '*result_ut1* 欠陥位置 母材部である baseWeight)))

(defun deposit_groove_haz_db (z d t gap depth angle)
;K形開先の欠陥位置の決定 (溶接金属部, 開先部, HAZ部)
  (let ((depositWeight (max (min (- 1 (* (- z (* (- depth d)(tan (radian angle)))
      gap -2.0)/(1 5.5)))
      (+ 1 (* (+ z -2.0)/(1 5.5)))
      1)
      0))
      (grooveWeight (min (max (min (+ 1 (* (- z (* (- depth d)(tan (radian angle)))
      gap -0.4)/(1 0.9)))
      (- 1 (* (- z (* (- depth d)(tan (radian angle)))
      gap 1.5)/(1 2.0)))
      (min (- 1 (* (+ z -0.4)/(1 0.9)))
      (+ 1 (* (+ z 1.5)/(1 2.0)))
      0)
      1))
      (rootWeight (max (min (+ 1 (* (- d depth -2)/(1 2.0)))
      (- 1 (* (- z (* (- depth d)(tan (radian angle)))
      gap 1.6)/(1 0.9)))
      (+ 1 (* (+ z 1.6)/(1 0.9)))
      1)
      0))
      (hazWeight (min (max (min (+ 1 (* (- z (* (- depth d)(tan (radian angle)))
      gap 2.0)/(1 3.0)))
      (- 1 (* (- z (* (- depth d)(tan (radian angle)))
      gap 4.0)/(1 2.0)))
      (min (- 1 (* (+ z 2.0)/(1 3.0)))
      (+ 1 (* (+ z 4.0)/(1 2.0)))
      0)
      1))
      (baseWeight (min (max (+ 1 (* (- z (* (- depth d)(tan (radian angle)))
      gap 4.0)/(1 3.0)))
      (- 1 (* (+ z 4.0)/(1 3.0)))
      0)
      1)))
      (b_push nil '*result_ut1* 欠陥位置 溶接金属部である depositWeight)
      (b_push nil '*result_ut1* 欠陥位置 開先部である grooveWeight)
      (b_push nil '*result_ut1* 欠陥位置 溶接金属ルート部である rootWeight)
      (b_push nil '*result_ut1* 欠陥位置 HAZ部である hazWeight)
      (b_push nil '*result_ut1* 欠陥位置 母材部である baseWeight)))

```

```

      1)))
      (b_push nil '*result_ut1* 欠陥位置 溶接金属部である depositWeight)
      (b_push nil '*result_ut1* 欠陥位置 開先部である grooveWeight)
      (b_push nil '*result_ut1* 欠陥位置 溶接金属ルート部である rootWeight)
      (b_push nil '*result_ut1* 欠陥位置 HAZ部である hazWeight)
      (b_push nil '*result_ut1* 欠陥位置 母材部である baseWeight)))

(defun data_t1 ()
;点数の計算
  (let ((length1 (search_value E高さ (H線基準) *input_ut2*))
      (length2 (search_value 指示長さ *input_ut2*))
      (length3 (search_value 掘り走査 *input_ut1*))
      (length4 (search_value 前後走査 *input_ut2*))
      (t 0))
      (cond ((<= length1 -6) (setq t (+ t 1)))
      ((< length1 0) (setq t (+ t 3)))
      (t (setq t (+ t 2))))
      (cond ((< length2 10) (setq t (+ t 1)))
      ((< length2 30) (setq t (+ t 3)))
      (t (setq t (+ t 2))))
      (cond ((equal length3 '8 d B未満) (setq t (+ t 1)))
      ((equal length3 '8 d B以上 15 d B以下) (setq t (+ t 2)))
      ((equal length3 '15 d Bを超える) (setq t (+ t 3)))
      (cond ((equal length4 '10 mm未満) (setq t (+ t 1)))
      ((equal length4 '10 mm以上 20 mm以下) (setq t (+ t 2)))
      ((equal length4 '20 mmを超える) (setq t (+ t 3)))
      (cond ((<= t 4) (b_push nil '*result_ut3* 点数 約4である 1.0))
      ((= t 5) (b_push nil '*result_ut3* 点数 約5である 1.0))
      ((= t 6) (b_push nil '*result_ut3* 点数 約6である 1.0))
      ((= t 7) (b_push nil '*result_ut3* 点数 約7である 1.0))
      ((= t 8) (b_push nil '*result_ut3* 点数 約8である 1.0))
      ((= t 9) (b_push nil '*result_ut3* 点数 約9である 1.0))
      ((= t 10) (b_push nil '*result_ut3* 点数 約10である 1.0))
      ((= t 11) (b_push nil '*result_ut3* 点数 約11である 1.0))
      ((= t 12) (b_push nil '*result_ut3* 点数 約12である 1.0))))))

(defun data_t1_t1 ()
;E高さと欠陥指示長さとの溶接材の板厚を欠陥の等級判定用のデータに変換
  (let ((echo (search_value E高さ (H線基準) *input_ut2*))
      (t (search_value 板厚 *input_w2*))
      (l (search_value 指示長さ *input_ut2*)))
      (echo_class echo)
      (board_t t)
      (f_length18 l)
      (f_length_t t l)
      (f_length60 l)))

(defun echo_class (echo)
;E高さと欠陥指示長さとの溶接材の板厚を欠陥の等級判定用のデータに変換
  (cond ((<= echo -6) (b_push nil '*result_ut2* 'E区分 'M線以下 1.0))
      ((< echo 0) (b_push nil '*result_ut2* 'E区分 'H線を超える 1.0))
      (t (b_push nil '*result_ut2* 'E区分 'M線を超えH線以下 1.0))))

(defun board_t (t)
;溶接材の板厚を欠陥の等級判定用のデータに変換
  (cond ((<= t 18) (b_push nil '*result_ut2* 板厚T 18以下である 1.0))
      ((<= t 60) (b_push nil '*result_ut2* 板厚T 60以上である 1.0))
      (t (b_push nil '*result_ut2* 板厚T 18を超え60未満である 1.0)))

(defun f_length18 (l)
;欠陥指示長さを欠陥の等級判定用のデータに変換
  (cond ((<= l 4) (b_push nil '*result_ut2* 欠陥指示長さ 4以下である 1.0))
      ((<= l 6) (b_push nil '*result_ut2* 欠陥指示長さ 6以下である 1.0))
      ((<= l 9) (b_push nil '*result_ut2* 欠陥指示長さ 9以下である 1.0))
      ((<= l 18) (b_push nil '*result_ut2* 欠陥指示長さ 18以下である 1.0))
      (t (b_push nil '*result_ut2* 欠陥指示長さ 18を超える 1.0)))

(defun f_length_t (t l)
  (cond ((<= l (/ t 4)) (b_push nil '*result_ut2* 欠陥指示長さ
      T/4以下である 1.0))
      ((<= l (/ t 3)) (b_push nil '*result_ut2* 欠陥指示長さ
      T/3以下である 1.0))
      ((<= l (/ t 2)) (b_push nil '*result_ut2* 欠陥指示長さ
      T/2以下である 1.0))
      ((<= l t) (b_push nil '*result_ut2* 欠陥指示長さ
      T以下である 1.0))
      (t (b_push nil '*result_ut2* 欠陥指示長さ
      Tを超える 1.0)))

(defun f_length60 (l)
  (cond ((<= l 15) (b_push nil '*result_ut2* 欠陥指示長さ 15以下である 1.0))
      ((<= l 20) (b_push nil '*result_ut2* 欠陥指示長さ 20以下である 1.0))
      ((<= l 30) (b_push nil '*result_ut2* 欠陥指示長さ 30以下である 1.0))
      ((<= l 60) (b_push nil '*result_ut2* 欠陥指示長さ 60以下である 1.0))
      (t (b_push nil '*result_ut2* 欠陥指示長さ 60を超える 1.0)))

(defun r10 (z)

```



```
;小数点以下1の位を四捨五入
(/ (round (* z 10)) 10.0))
```

```
(defun r100 (z)
;小数点以下2の位を四捨五入
(/ (round (* z 100)) 100.0))
```

```
(defun radian (angle)
;度からラジアンへの変換
(/ (* angle (pi)) 180))
```

```
;-----[ method for flaw_estimation ]-----
```

```
;超音波探傷試験データより欠陥の寸法を計算する
```

```
(defun estimation ()
  (let ((flaw_f (fget '*whylog_ef* 'rule_name 'value))
        (echo (search_value E高さ (H線基準) *input_ut2*))
        (frequency (search_value 周波数 *input_ut2*))
        (beam_l (search_value B路程 *input_ut2*))
        (flaw_l (search_value 指示長さ *input_ut2*))
        (angle (search_value 屈折角 *input_ut2*))
        (metal (search_value 材料 *input_w1))
        (velocity (cond ((equal metal S S鋼) 3230)
                          ((equal metal S M鋼) 3230)
                          (t 3230))))
    (let ((echo_r (expt 1.122 echo)))
      (do ((flaw_f1 flaw_f (cdr flaw_f1))
          (rule_list (fget 'figure_rules 'rules 'value))
          ((null flaw_f1))
          (let ((flaw (estimation_aux1 (car flaw_f1) rule_list)))
            (cond ((equal flaw 円形平面欠陥)
                   (td_root echo_r frequency velocity beam_l
                             angle (car flaw_f1)))
                  ((equal flaw 帯状平面欠陥)
                   (tl_root echo_r frequency velocity beam_l
                             angle (car flaw_f1)))
                  ((equal flaw 横穴欠陥)
                   (th_root echo_r frequency velocity beam_l flaw_l
                             angle (car flaw_f1)))
                  ((equal flaw 球状欠陥)
                   (ts_root echo_r frequency velocity beam_l
                             angle (car flaw_f1)))
                  ((equal flaw 縦穴欠陥)
                   (tv_root echo_r frequency velocity beam_l
                             angle (car flaw_f1)))
                  ((equal flaw 直線溝欠陥)
                   (tn_root echo_r frequency velocity beam_l
                             angle (car flaw_f1))))))))))
```

```
(defun estimation_aux1 (rule_name rule_list)
  (cond ((null rule_list)
        (prtsen '(プログラムにバグの可能性有り))
        ((equal rule_name (cadadr rule_list))
         (caddr (cadadr (cddr rule_list))))
        (t (estimation_aux1 rule_name (cdr rule_list)))))
```

```
(defun td_root (echo_r frequency velocity beam_l angle flaw_f1)
  (let ((length (td echo_r frequency velocity beam_l angle)))
    (cond ((null length)
           (fput '*whylog_es* (cadr (cadadr (get flaw_f1 'frame)))
                 '欠陥寸法 限界寸法以上である)
           (put '*b_board* '*result_e*
                '(((欠陥寸法 は 限界寸法 以上である))))
           (t (fput '*whylog_es* (cadr (cadadr (get flaw_f1 'frame)))
                    '欠陥寸法 length)
              (put '*b_board* '*result_e*
                   (list (list (list '欠陥寸法 'は length))))))))))
```

```
(defun tl_root (echo_r frequency verosity beam_l angle flaw_f1)
  (let ((length (tl echo_r frequency verosity beam_l angle)))
    (cond ((null length)
           (fput '*whylog_es* (cadr (cadadr (get flaw_f1 'frame)))
                 '欠陥寸法 限界寸法以上である)
           (put '*b_board* '*result_e*
                '(((欠陥寸法 は 限界寸法 以上である))))
           (t (fput '*whylog_es* (cadr (cadadr (get flaw_f1 'frame)))
                    '欠陥寸法 length)
              (put '*b_board* '*result_e*
                   (list (list (list '欠陥寸法 'は length))))))))))
```

```
(defun th_root (echo_r frequency verosity beam_l flaw_l angle flaw_f1)
  (let ((length (th echo_r frequency verosity beam_l flaw_l angle)))
    (fput '*whylog_es* (cadr (cadadr (get flaw_f1 'frame)))
          '欠陥寸法 length)
    (put '*b_board* '*result_e*
         (list (list (list '欠陥寸法 'は length))))))
```

```
(defun ts_root (echo_r frequency verosity beam_l angle flaw_f1)
```

```
(let ((length (ts echo_r frequency verosity beam_l angle)))
  (fput '*whylog_es* (cadr (cadadr (get flaw_f1 'frame)))
        '欠陥寸法 length)
  (put '*b_board* '*result_e*
       (list (list (list '欠陥寸法 'は length))))))
```

```
(defun tv_root (echo_r frequency verosity beam_l angle flaw_f1)
  (let ((length (tv echo_r frequency verosity beam_l angle)))
    (fput '*whylog_es* (cadr (cadadr (get flaw_f1 'frame)))
          '欠陥寸法 length)
    (put '*b_board* '*result_e*
         (list (list (list '欠陥寸法 'は length))))))
```

```
(defun tn_root (echo_r frequency verosity beam_l angle flaw_f1)
  (let ((length (tn echo_r frequency verosity beam_l angle)))
    (cond ((null length)
           (fput '*whylog_es* (cadr (cadadr (get flaw_f1 'frame)))
                 '欠陥寸法 限界寸法以上である)
           (put '*b_board* '*result_e*
                '(((欠陥寸法 は 限界寸法 以上である))))
           (t (fput '*whylog_es* (cadr (cadadr (get flaw_f1 'frame)))
                    '欠陥寸法 length)
              (put '*b_board* '*result_e*
                   (list (list (list '欠陥寸法 'は length))))))))))
```

```
;欠陥寸法の計算
```

```
(defun td (e f v l angle)
;円形平面欠陥の欠陥寸法の計算
;限界寸法以上のときはnilを返す
  (let ((rgf (/ 1 2))
        (rgs (r_g_s f v l angle))
        (rif 1)
        (ris (r_l angle)))
    (cond ((< e (e_ratio rgf rgs rif ris))
           (sqrt (/ (* e rgs 4 ris (wave_l f v) l)
                    (* rif (pi))))
           (t nil))))
```

```
(defun tl (e f v l angle)
;帯状平面欠陥の欠陥寸法の計算
;限界寸法以上のときはnilを返す
  (let ((rgf (/ 1 2))
        (rgs (r_g_s f v l angle))
        (rif 1)
        (ris (r_l angle)))
    (cond ((< e (e_ratio rgf rgs rif ris))
           (let ((a (/ (* 2 l (wave_l f v)) 100)))
             (let ((b (round (* (sqrt a) 1000 e rgs ris))))
               (* 1.0 (/ b 100))))
           (t nil))))
```

```
(defun th (e f v l tl angle)
;横穴欠陥の欠陥寸法の計算
;限界寸法以上のときも欠陥寸法を返す
  (let ((tcr (tcr_cp_hh f v l))
        (rgs (r_g_s f v l angle))
        (rif 1)
        (ris (r_l angle)))
    (cond ((< tl tcr)
           (/ (* e e rgs rgs 16 l l ris (wave_l f v))
              (* rif (pi) tl tl)))
          (t (/ (* e e rgs rgs ris ris 8 l)
                 (* rif rif))))))
```

```
(defun ts (e f v l angle)
;球状欠陥の欠陥寸法の計算
;限界寸法は存在しないので限界寸法以上も以下も同じである
  (let ((rgs (r_g_s f v l angle))
        (rif 1)
        (ris (r_l angle)))
    (* 1.0 (/ (round (* 100 (/ (* e rgs ris 4 l)
                                rif)))
              100))))
```

```
(defun tv (e f v l angle)
;縦穴欠陥の欠陥寸法の計算
;限界寸法以上のときも欠陥寸法を返す
  (let ((tcr (tcr_vh f v l angle))
        (rgf (r_g_cr_vh tcr f v l angle))
        (rgs (r_g_s f v l angle))
        (rif (r_l angle))
        (ris (r_l angle)))
    (cond ((< e (e_ratio rgf rgs rif ris))
           (expt (/ (* e e rgs rgs ris ris 4 l l (wave_l f v))
                    (* rif rif (pi) (sin (radian angle))))
                 (/ 1 3)))
          (t (/ (* e e rgs rgs ris ris 8 l (sin (radian angle)))
                 (* rif rif))))))
```



```
(defun tn (e f v l angle)
;直線溝欠陥の欠陥寸法の計算
;限界寸法以上の場合はnilを返す
(let ((rgf (/ 1 2))
      (rgs (r_g_s f v l angle))
      (rif (r_i angle))
      (ris (r_i angle)))
  (cond ((< e (e_ratio rgf rgs rif ris))
        (/ (* e rgs ris (sqrt (/ 1 (wave_l f v) 2)))
            (* rif (sin (radian angle)))))
        (t nil))))
```

;形状反射率の計算

```
(defun r_g_cp (t f v l)
;円形平面欠陥の形状反射能率の計算
(/ (* t t (pi))
   (* 4 l (wave_l f v))))
```

```
(defun r_g_sp (t f v l)
;帯状平面欠陥の形状反射能率の計算
(sqrt (/ (* t t)
          (* 2 l (wave_l f v))))
```

```
(defun r_g_hh (t f v l tl)
;横穴欠陥の形状反射能率の計算
;欠陥の大きさが限界寸法以下の場合)
(sqrt (/ (* t tl tl (pi))
          (* 16 l l (wave_l f v))))
```

```
(defun r_g_cr_hh (t f v l)
;横穴欠陥の場合の形状反射能率の計算
;欠陥の大きさが限界寸法以上の場合)
(sqrt (/ t (* 8 l)))
```

```
(defun r_g_sh (t l)
;球状欠陥の形状反射率の計算
;欠陥の大きさが限界寸法以下の場合)
(/ t (* 4 l))
```

```
(defun r_g_cr_sh (t l)
;球状欠陥の形状反射率の計算
;欠陥の大きさが限界寸法以上の場合)
(/ t (* 4 l))
```

```
(defun r_g_vh (t f v l angle)
;縦穴欠陥の形状反射率の計算
;欠陥の大きさが限界寸法以下の場合)
(sqrt (/ (* t t t (pi) (sin (radian angle)))
          (* 4 l l (wave_l f v))))
```

```
(defun r_g_cr_vh (t f v l angle)
;縦穴欠陥の形状反射能率の計算
;欠陥の大きさが限界寸法以上の場合)
(let ((a (* (/ t 8 l) 100))
      (b (/ 1 (sin (radian angle)))))
  (/ (sqrt (* a b)) 10))
```

```
(defun r_g_lg (t f v l angle)
;直線溝欠陥の形状反射率の計算
(* (sqrt (/ (* 2 t t) (* l (wave_l f v))))
   (sin (radian angle)))
```

;限界寸法の計算

```
(defun tcr_cp_hh (f v l)
;円形平面欠陥，横穴欠陥の限界寸法の計算
(sqrt (/ (* 2 l (wave_l f v))
          (pi)))
```

```
(defun tcr_sp (f v l)
;帯状平面欠陥の限界寸法の計算
(sqrt (/ (* l (wave_l f v))
          2 ))
```

```
(defun tcr_vh (f v l angle)
;縦穴平面欠陥の限界寸法の計算
(/ (sqrt (/ (* l (wave_l f v)) (* 2 (pi))))
   (sin (radian angle)))
```

```
(defun tcr_lg (f v l angle)
;直線溝欠陥の限界寸法の計算
(/ (sqrt (/ (* l (wave_l f v)) 8))
   (sin (radian angle)))
```

;標準欠陥の形状反射能率の計算

```
(defun r_g_s (f v l angle)
```

```
(let ((beam_l (/ 15 (cos (radian (* 10 (round (/ angle 10)))))))
  (let ((tcr (tcr_vh f v beam_l angle)))
    (cond ((< 4 tcr) ;標準欠陥寸法が限界寸法以下の場合
           (r_g_vh 4 f v beam_l angle))
          (t ;標準欠陥寸法が限界寸法以上の場合
             (r_g_cr_vh 4 f v beam_l angle)))))
```

;欠陥エコー高さと標準エコー高さの比の計算

```
(defun e_ratio (rgf rgs rif ris)
(/ (* rgf rif)
   (* rgs ris))
```

;界面反射能率の計算

```
(defun r_i (angle)
(let ((a (round angle)))
  (cond ((equal a 68) 0.5) ;屈折角が約70°の場合
        ((equal a 69) 0.5)
        ((equal a 70) 0.5)
        ((equal a 71) 0.5)
        ((equal a 72) 0.5)
        ((equal a 43) 0.5) ;屈折角が約45°の場合
        ((equal a 44) 0.5)
        ((equal a 45) 0.5)
        ((equal a 46) 0.5)
        ((equal a 47) 0.5)
        (t 1))))
```

```
(defun wave_l (f v)
;音速と周波数より波長を計算する
(/ v (* f 1000)))
```

---[ method for display\_e ]-----

;欠陥種類，欠陥の寸法および欠陥の等級を表示する

```
(defun display_e ()
(clear-screen)
(set-cursor)
(display_e2))
```

```
(defun display_e2 ()
;推論結果およびデータを表示する
(let ((data1 (get '*b_board* 'input_w1*))
      (data2 (get '*b_board* 'input_w2*))
      (data3 (get '*b_board* 'input_ut2*))
      (data4 (get '*b_board* 'input_ut1*))
      (data5 (get '*b_board* 'input_ut*))
      (data_list (fget_v_d curframe 'data_name))
      (result_1 (get '*b_board* 'result_ut*))
      (result_2 (get '*b_board* 'result_ec*))
      (result_list (fget_v_d curframe 'result_name)))
  (terpri 1)
  (prtsen '(溶接条件データを表示します))
  (terpri 1)
  (display_e_aux (append data1 data2) data_list)
  (terpri 1)
  (y-or-n-p-2 "次に超音波探傷試験データを表示します。[Y]を押して下さい。")
  (clear-screen)
  (set-cursor)
  (prtsen '(超音波探傷試験データを表示します))
  (terpri 1)
  (display_e_aux (append data3 data4) data_list)
  (terpri 1)
  (prtsen '(欠陥位置))
  (display_e_aux (append data5) data_list)
  (terpri 1)
  (y-or-n-p-2 "次に推論結果(*)を表示します。[Y]を押して下さい。")
  (clear-screen)
  (set-cursor)
  (prtsen '(推論結果))
  (terpri 1)
  (prtsen '(欠陥種類))
  (display_result_ut result_1 result_list)
  (terpri 1)
  (prtsen '(欠陥等級))
  (display_result_e result_2 result_list)))
```

```
(defun display_e_aux (data data_list)
;dataリストに基づいてデータを表示する
(cond ((null data) t)
      ((member (caadar data) data_list 'equal)
       (prtsen (append '( ) (cadar data)))
       (display_e_aux (cdr data) data_list))
      (t (display_e_aux (cdr data) data_list))))
```

```
(defun display_result_ut (result result_name)
```



```

;欠陥種類の推論結果を表示する
(cond ((null result) (prtsen '(欠陥種類の推論結果がありません)))
      (t (do ((result2 result (cdr result2)))
              ((null result2))
              (cond ((member (caadar result2) result_name 'equal)
                     (prtsen (list '* (cadar result2) 'の確信度は
                                   (cond ((null(cddar result2)) 1.0)
                                         (t (caddar result2)) )
                                   'である))
                     (cond ((equal (caadar result2) '欠陥)
                            (let ((aaa (display_result_e_aux1 (cadar result2)
                                                                (fget 'ut_rules 'rules 'value))))
                              (let ((bbb (car (build_facets '*whylog-es* aaa))))
                                (cond ((null bbb)
                                       (t (prtsen (list '
                                                         (car bbb)
                                                         'は
                                                         (cadr bbb)
                                                         'm m)) ))))))
                            (let ((bbb (car (build_facets '*whylog-es* aaa))))
                              (cond ((null bbb)
                                     (t (prtsen (list '
                                                         (car bbb)
                                                         'は
                                                         (cadr bbb)
                                                         'm m)) ))))))
                            (t (prtsen (list '
                                              (car bbb)
                                              'は
                                              (cadr bbb)
                                              'm m)) )))))))))

(defun display_result_e_aux1 (result rule_list)
  (cond ((null rule_list))
        ((and (equal result (caadar (cdddar rule_list)))
              (member (cadar rule_list) (cdr (build_facets '*whylog-ut* 'rule_name)) 'equal))
         (cadar rule_list))
        (t (display_result_e_aux1 result (cdr rule_list)))))

(defun sum_calculation (result)
  (do ((result2 result (cdr result2))
      (sum 0))
      ((null result2) sum)
      (setq sum (+ sum (caddar result2))) ) )

(defun display_result_e (result result_name)
;欠陥の等級を表示する
  (cond ((null result) (prtsen '(欠陥の等級の推論結果がありません)))
        (t (do ((result2 result (cdr result2)))
                ((null result2))
                (cond ((member (caadar result2) result_name 'equal)
                       (prtsen (list '* (cadar result2) 'の確信度は
                                       (cond ((null(cddar result2)) 1.0)
                                             (t (caddar result2)) )
                                       'である)) ))))))))

;-----[ method for msg_e ]-----

(defun msg_e ()
  (terpri)
  (cond ((y-or-n-p "推論に関して説明が必要ですか")
        (how_why))
        (t (write-string "以上でシステムを終了します")
            (t)))

;-----[ how_why ]-----

(defun how_why ()
  (let ((frame1 (car (fget '*whylog-ut* 'frame 'value)))
        (facts1 (get '*b_board* '*result_ut*))
        (frame2 (car (fget '*whylog-ec* 'frame 'value)))
        (facts2 (get '*b_board* '*result_ec*)))
    (cond ((null (usedRuleList 1 facts1 (fget frame1 'rules 'value) nil frame1))
           (clear-screen)
           (set-cursor)
           (terpri)
           (prtsen '(最後にデータおよび推論結果を表示してシステムを終了します))
           (display_e2))
          ((null (usedRuleList 2 facts2 (fget frame2 'rules 'value) nil frame2))
           (clear-screen)
           (set-cursor)
           (terpri)
           (prtsen '(最後にデータおよび推論結果を表示してシステムを終了します))
           (display_e2))
          (t (terpri)
              (cond ((y-or-n-p "推論過程に関する説明は以上で終わります。もう一度説明が必要ですか。
                              (how_why))
                    (t (clear-screen)
                        (set-cursor)
                        (prtsen '(最後にデータおよび推論結果を表示してシステムを終了します))
                        (display_e2)))))))

(defun usedRuleList (no facts rule_list possibilities frame)
  (clear-screen)
  (set-cursor)
  (cond ((null facts)
        (cond ((eq no 1) (prtsen '(欠陥種類の推定結果が存在しません)))
              (t (prtsen '(欠陥の等級の推定結果が存在しません))))
        (t (usedRuleListAux (sum_calculation facts)
                             facts rule_list possibilities frame))))

```

```

facts rule_list possibilities frame))))

(defun usedRuleListAux (sum facts rule_list possibilities frame)
  (clear-screen)
  (set-cursor)
  (cond ((null facts)
        (t (terpri)
            (prtsen '(結論を導いたルールを示します))
            (setq possibilities (usedRuleListAux1 (car facts) rule_list nil frame))
            (how sum (cdr facts) possibilities)
            (terpri 1)
            (cond ((y-or-n-p "推論過程の説明を続けますか")
                  (usedRuleListAux sum (cdr facts) rule_list nil frame))
                  (t nil))))))

(defun usedRuleListAux1 (fact rule_list possibilities frame)
  (cond ((null rule_list)
        (prtsen '(プログラムにバグの可能性あり))
        possibilities)
        ((equal (car fact) (cadar rule_list))
         (push (car rule_list) possibilities)
         (usedRuleListAux2 (build_facets (cadar rule_list) 'ruleChain) possibilities))
        (t (usedRuleListAux1 fact (cdr rule_list) possibilities frame))))

(defun usedRuleListAux2 (facets possibilities)
  (cond ((null facets) possibilities)
        ((null (get (caar facets) 'frame))
         (push (car facets) possibilities)
         (usedRuleListAux2 (cdr facets) possibilities))
        (t (setq possibilities
                (usedRuleListAux3 (car facets)
                                  (fget (caar facets) 'rules 'value)
                                  possibilities
                                  (caar facets))))
            (usedRuleListAux2 (cdr facets) possibilities))))

(defun usedRuleListAux3 (rule_name rule_list possibilities frame)
  (cond ((null rule_name) possibilities)
        (t (setq possibilities
                (usedRuleListAux4 (car rule_name) rule_list possibilities frame))
            (usedRuleListAux3 (cdr rule_name) rule_list possibilities frame))))

(defun usedRuleListAux4 (rule_name rule_list possibilities frame)
  (cond ((null rule_list)
        (prtsen '(プログラムにバグの可能性あり))
        possibilities)
        ((equal rule_name (cadar rule_list))
         (push (car rule_list) possibilities)
         (usedRuleListAux2 (build_facets rule_name 'ruleChain) possibilities))
        (t (usedRuleListAux4 rule_name (cdr rule_list) possibilities frame))))

;-----[ how ]-----

;推論過程に関する説明の表示

(defun how (sum fact possibilities)
  (cond ((null possibilities)
        (prtsen '(プログラムにバグの可能性あり))
        (t (how_aux2 sum fact)
            (how_aux3 possibilities))))

(defun how_aux2 (sum fact)
  (cond ((equal 6 (length (car fact)))
        (terpri 1)
        (prtsen (list (car (car fact))
                      (cadr (car fact))
                      (caddr (car fact)) ))
        (prtsen (list (caddar (car fact))
                      (caddar (cdr (car fact)))
                      (caddar (cddr (car fact)))) ))
        (t (terpri)
            (prtsen (car fact))))
  (prtsen (list ' の確信度は (cadr fact) 'である))
  (terpri 1)
  (y-or-n-p-2 "以下に上記の結論を導いたルールを表示します。[Y]を押して下さい。"))

(defun how_aux3 (possibilities)
  (cond ((null possibilities)
        (t (cond ((equal (caar possibilities) 'rule)
                  (terpri 1)
                  (prtsen (list '(ルール) (cadar possibilities)))
                  (prtif (cdr (caddar possibilities)))
                  (prtthen (cadr (cdddar possibilities))))
                ((or (equal (caar possibilities) '*input_w1*
                          (equal (caar possibilities) '*input_w2*
                                (equal (caar possibilities) '*input_ut1*

```



```

(equal (caar possibilities) '*input_ut2*))
(terpri 1)
(prtsen '(入力データ))
(do ((list (cdar possibilities) (cdr list)))
    ((null list))
    (prtsen (append '(
                        (list (caar list))
                        '(の確信度は)
                        (cdar list))))))
(t (terpri 1)
    (prtsen '(データ))
    (do ((list (cdar possibilities) (cdr list)))
        ((null list))
        (prtsen (append '(
                            (list (caar list))
                            '(の確信度は)
                            (cdar list))))))
(terpri 1)
(y-or-n-p-2 " [Y] を押すと次に進みます。")
(how_aux3 (cdr possibilities))))

(defun prtif (list)
  (prtsen (append '( i f ) (car list)))
  (prtIfAux (cdr list)))

(defun prtthen (list)
  (prtsen (append '( t h e n ) (list (caar list)) '(の確信度は)
                  (cond ((null (cdar list)) '(1))
                        (t (cdar list))))))
  (prtThenAux (cdr list)))

(defun prtIfAux (list)
  (cond ((null list))
        (t (prtsen (append '(
                            ) (car list)))
            (prtIfAux (cdr list))))))

(defun prtThenAux (list)
  (cond ((null list))
        (t (prtsen (append '(
                            ) (list (caar list)) '(の確信度は)
                            (cond ((null (cdar list)) '(1))
                                (t (cdar list))))))
            (prtThenAux (cdr list))))))

(rds b:control)

```

```

**** mulisp86 version ***** 92.11.14 *****
:
: File name : FRAME. LSP
: Feature   : フレーム・システム
:
: *****

(defun fint (frame frame_list)
;フレームの格納
  (put frame 'frame frame_list))

(defun fdisp (frame)
;フレームの表示
  (get frame 'frame))

(defun fget (frame slot facet)
;与えられたフレーム, スロット, ファセットから値を取り出す
  (cdr (assoc facet
              (cdr (assoc slot
                          (cdr (get frame 'frame)))))))

(defun build_facets (frame slot)
;与えられたフレーム, スロットから値を取り出す
  (cdr (assoc slot
              (cdr (get frame 'frame)))))

(defun fput (frame slot facet value)
;与えられたフレーム, スロット, ファセットに値を格納する
  (cond ((null value))
        (t (cond ((not (member frame $frames))
                    (push frame $frames)))
            (let ((value_list (follow_path (list slot facet)
                                             (fget_frame frame))))
              (cond ((member value value_list) nil)
                    (t (rplacd (last value_list) (list value)
                                value)))))))

(defun follow_path (path a_list)
  (cond ((null path) a_list)
        (t (follow_path (cdr path) (extend (car path) a_list)))))

(defun extend (key a_list)
  (cond ((assoc key (cdr a_list)))
        (t (cadr (rplacd (last a_list) (list (list key)))))))

(defun last (a_list)
  (cond ((null a_list) nil)
        (t (last_aux a_list))))

(defun last_aux (a_list)
  (cond ((null (cdr a_list)) a_list)
        (t (last_aux (cdr a_list)))))

(defun fget_frame (frame)
  (cond ((get frame 'frame))
        (t (self (get frame 'frame) (list frame)))))

(defun fremove (frame slot facet value)
;与えられたフレーム, スロット, ファセットの値を消去する
  (let ((value_list (follow_path (list slot facet)
                                  (fget_frame frame))))
    (cond ((member value value_list)
            (delete value value_list)
            t)
          (t nil))))

(defun fdelete (frame)
;フレームの削除
  (remprop frame 'frame))

(defun fcheck (frame slot facet value)
;与えられたフレーム, スロット, ファセットに値があるかどうかをチェックする
  (cond ((member value (fget_frame slot facet)) t)
        (t nil)))

(defun fclamp (frame1 frame2 slot)
;与えられた2つのフレームをくつつける
  (rplacd (fget_frame frame1)
          (list (follow_path (list slot) (fget_frame frame2))))
  slot)

(defun fget_i (frame slot)
;与えられたフレーム, スロットから値を取り出す
;与えられたフレームに値がない場合には a k o スロットが示すフレームから検索する

```



```

(fget_il (fget_classes frame) slot))

(defun fget_il (frames slot)
  (cond ((null frames) nil)
        ((fget (car frames) slot 'value))
        (t (fget_il (cdr frames) slot))))

(defun fget_classes (start)
  (reverse (fget_classes1 (list start) nil)))

(defun fget_classes1 (queue classes)
  (cond ((null queue) classes)
        (t (fget_classes1
              (append (fget (car queue) 'ako 'value)
                      (cdr queue))
              (cond ((member (car queue) classes)
                    classes)
                    (t (cons (car queue) classes))))))))

(defun fget_if (frame slot facet)
  ;ファセット名と継承により値を得る
  ;マスターフレームのvalueと異なるファセットから値を得るためのチェックを行う
  (fget_if_aux (fget_classes frame)))

(defun fget_if_aux (classes)
  (cond ((null classes) nil)
        ((setq result (fget (car classes) slot facet)) result)
        (t (fget_if_aux (cdr classes)))))

(defun fget_v_d (frame slot)
  ;valueファセットまたはdefaultファセットから値を得る
  (cond ((fget frame slot 'value))
        ((fget frame slot 'default))))

(defun fget_c (slot)
  (fget_v_d curframe slot))

(defun ask (frame slot)
  (print (append '(please supply a value for the)
                 (list slot)
                 '(slot in the)
                 (list frame)
                 '(frame))))
  (terpri)
  (let ((response (read)))
    (cond (response (list response))
          (t nil))))

(defun fget_n (frame slot)
  ;継承とデフォルトを利用して値を得る
  (let ((classes (fget_classes frame)))
    (cond ((fget_n1 slot classes 'value))
          ((fget_n1 slot classes 'default))
          ((fget_n2 slot classes 'if_needed))
          (t nil))))

(defun fget_n1 (slot classes key)
  (cond ((null classes) nil)
        ((fget (car classes) slot key))
        (t (fget_n1 slot (cdr classes) key))))

(defun fget_n2 (slot classes key)
  (cond ((null classes) nil)
        ((mapcan '(lambda (e) (apply e nil))
                  (fget (car classes) slot key)))
        (t (fget_n2 slot (cdr classes) key))))

(defun fput_p (frame slot facet value)
  ;与えられたフレーム、スロット、ファセットに値を格納する
  (cond ((fput frame slot facet value)
        (mapcar '(lambda (e)
                    (mapcar 'funcall
                            (fget e slot 'if_added)))
                  (fget_classes frame)
                  value)))

(defun store_frames (framelist)
  ;フレームをメモリーに格納する
  (cond ((null framelist) t)
        (t (let ((frame (car framelist))
                  (name (caar framelist)))
              (cond ((not (atom name))
                    (prtsen '(not named by atom))

```

```

(store_frames (cdr framelist)))
(t (put name 'frame frame)
  ;フレームがマスターフレームであれば、
  ;masterframesへ名前を格納する
  (cond ((equal (fget name 'ako 'value) '(master))
        (or (member name *masterframes*)
            (setq *masterframes*
                  (cons name *masterframes*)))
        (store_frames (cdr framelist)))
    ;フレームがマスターフレーム以外なら、
    ;framesへ名前を格納する
    (t (or (member name *frames*)
          (setq *frames* (cons name *frames*)))
        (store_frames (cdr framelist))))))

(defun freplace (frame slot facet value)
  ;与えられたフレーム、スロット、ファセットの値を新しい値に入れ換える
  ;最初に2つ以上の値があるときは不適当である
  (replacd (fassoc facet
                   (fassoc slot
                           (fget_frame frame)))
           (list (list value))))

(defun fassoc (key a_list)
  (cond ((assoc key (cdr a_list))
        (t (cdr (rplacd (last a_list)
                        (list (list key)))))))

(defun build_slot_list (frame)
  ;フレームに格納されているスロットのリストを作成する
  (mapcar 'car (cdr (get frame 'frame))))

(rds b:method)

```



```

**** mulisp86 version ***** 92.11.14 *****
;
; File name : RULE.LSP
; Feature   : 推論エンジンのためのルールベースシステム
;
;*****

;---( データの入力 )-----
(defun input_assertions (input)
  ; *data* を assertions にコピーする
  (cond ((null input))
        (t (setq assertions (append (get '*b_board* (car input)) assertions)
            (input_assertions (cdr input))))))

;---( 前向き推論 )-----
(defun forward_chain (rules)
  (setq assertions nil)
  (input_assertions input)
  (cond ((null (int_variable))
        (forward_chain_aux rules))))

(defun forward_chain_aux (rules_to_try)
  (cond ((null rules_to_try) progress_made)
        (t (use_rule (car rules_to_try))
            (forward_chain_aux (cdr rules_to_try))
            (t (forward_chain_aux (cdr rules_to_try))))))

(defun int_variable ()
  ;変数の初期化
  (progl (put '*b_board* result nil)
         (fputWhylog 'forward forward)
         (fputWhylog 'frame (list curframe))
         (setq progress_made nil)))

(defun fputWhylog (slot value)
  (cond ((null value))
        (t (fput rules_used slot 'value (car value))
            (fputWhylog slot (cdr value))))))

;---( ルールの処理 )-----
(defun use_rule (rule)
  (setq matchRatio 1.0)
  (setq rule_name (cadr rule))
  (setq if_list (reverse (cdr (caddr rule)))) ; i f 部分の処理
  (setq then_list (cdr (caddr rule))) ; t h e n 部分の処理
  (setq a_list_stream (cascade_through_patterns
    if_list
    (add_to_stream nil (make_empty_stream))))
  (setq action_stream (feed_to_actions rule_name
    then_list
    a_list_stream))
  (cond ((not (empty_stream_p action_stream))
        (feed_to_rules_used rule_name (reverse if_list) then_list a_list_stream)
        t)))

;---( i f 部分の処理 )-----
(defun cascade_through_patterns (patterns a_list_stream)
  ; i f 部分の条件を順番に処理する
  (cond ((null patterns) a_list_stream)
        (t (filter_a_list_stream (car patterns)
            (cascade_through_patterns (cdr patterns)
            a_list_stream))))))

(defun filter_a_list_stream (pattern a_list_stream)
  ; filter_assertionsへデータを送り込む
  (cond ((empty_stream_p a_list_stream) (make_empty_stream))
        (t (combine_streams
            (filter_assertions pattern (first_of_stream a_list_stream))
            (filter_a_list_stream pattern (rest_of_stream a_list_stream))))))

(defun filter_assertions (pattern initial_a_list)
  ;条件と assertions を照合する
  (do ((assertions assertions (cdr assertions))
      (a_list_stream (make_empty_stream)))
      ((or (null assertions) (not (null a_list_stream))) a_list_stream)
      (cond ((search_name (caar pattern) (fget 'membership 'name 'value))
            (let ((new_a_list
                  (match1 (car pattern) (cadr assertions) initial_a_list)
                  (matchRatio (matchRatioAux (car assertions))))
              (cond ((and new_a_list (>= new_a_list 0.01))
                    (setq a_list_stream
                          (add_to_stream new_a_list a_list_stream))
                    (cond ((not (numberp matchRatio))
                          (setq matchRatio 1.0))
                          (t (setq matchRatio
                                (* matchRatio new_a_list))))))
              (let ((new_a_list
                    (match (car pattern) (cadr assertions) initial_a_list)
                    (matchRatio (matchRatioAux (car assertions))))
                  (cond (new_a_list
                        (setq a_list_stream
                              (add_to_stream new_a_list a_list_stream))
                        (cond ((not (numberp matchRatio))
                              (setq matchRatio 1.0))
                              (t (setq matchRatio
                                    (* matchRatio matchRatio))))))
                  (defun matchRatioAux (assertions)
                    ;
                    (cond ((null (cdr assertions)) (car assertions))
                          (t (matchRatioAux (cdr assertions))))))
                  ;---( t h e n 部分の処理 )-----
                  (defun feed_to_actions (rule_name actions a_list_stream)
                    ;結果のリストをまとめて1つのリストにする
                    (cond ((empty_stream_p a_list_stream) (make_empty_stream))
                          (t (combine_streams
                              (spread_through_actions rule_name
                                actions
                                (first_of_stream a_list_stream))
                              (feed_to_actions rule_name
                                actions
                                (rest_of_stream a_list_stream))))))
                  (defun spread_through_actions (rule_name actions a_list)
                    ;照合が成功した結果のリストを作成する
                    (do ((actions actions (cdr actions))
                        (action_stream (make_empty_stream)))
                        ((null actions) action_stream)
                        (let ((new (newMatch (car actions)))
                            (setf (get '*b_board* result)
                                (remember (get '*b_board* result) rule_name
                                (car new)
                                (cadr new))))
                          (setf action_stream (add_to_stream new action_stream))))))
                  (defun newMatch (action)
                    ;確信度を計算しリストを作成する
                    (do ((aaa nil)
                        (action action (cdr action)))
                        ((null action) aaa)
                        (setq aaa (append aaa (newMatchAux action))))))
                  (defun newMatchAux (action)
                    ;確信度の計算
                    (cond ((null (cdr action))
                          (cond ((numberp (car action)) (list (* (car action) matchRatio)))
                                (t (append (list (car action)) (list matchRatio)))))
                          (t (list (car action)))))
                  (defun remember (p x y z)
                    ;確信度の大きい順に格納する
                    (cond ((< z 0.01)
                          p)
                          ((null p)
                          (cons (cons x (cons y (cons (r100 z) nil))) p))
                          ((equal y (cadr p))
                          (cond ((> z (caddr p))
                                (rplaca (car p) x)
                                (rplacd (cadr p) (cons (r100 z) nil))
                                p)
                                (t p)))
                          ((> z (caddr p))
                          (push (cons x (cons y (cons (r100 z) nil))) p)
                          (del_list y p)
                          p)
                          (t (cons (car p) (remember (cdr p) x y z))))))
                  (defun del_list (y p)
                    ;リストからの削除
                    (cond ((null (cdr p)) nil)
                          ((equal (caddr p) y) (rplacd p (caddr p)))
                          (t (del_list y (cdr p))))))
                  ;---( 利用したルールのリストを生成する )-----
                  (defun feed_to_rules_used (rule_name if_list then_list a_list_stream)

```

```

                    (add_to_stream new_a_list a_list_stream))
                    (cond ((not (numberp matchRatio))
                          (setq matchRatio 1.0))
                          (t (setq matchRatio
                                (* matchRatio new_a_list))))))
                    (* matchRatio matchRatio))))))
                    (defun matchRatioAux (assertions)
                      ;
                      (cond ((null (cdr assertions)) (car assertions))
                            (t (matchRatioAux (cdr assertions))))))
                    ;---( t h e n 部分の処理 )-----
                    (defun feed_to_actions (rule_name actions a_list_stream)
                      ;結果のリストをまとめて1つのリストにする
                      (cond ((empty_stream_p a_list_stream) (make_empty_stream))
                            (t (combine_streams
                                (spread_through_actions rule_name
                                  actions
                                  (first_of_stream a_list_stream))
                                (feed_to_actions rule_name
                                  actions
                                  (rest_of_stream a_list_stream))))))
                      (defun spread_through_actions (rule_name actions a_list)
                        ;照合が成功した結果のリストを作成する
                        (do ((actions actions (cdr actions))
                            (action_stream (make_empty_stream)))
                            ((null actions) action_stream)
                            (let ((new (newMatch (car actions)))
                                (setf (get '*b_board* result)
                                    (remember (get '*b_board* result) rule_name
                                    (car new)
                                    (cadr new))))
                              (setf action_stream (add_to_stream new action_stream))))))
                      (defun newMatch (action)
                        ;確信度を計算しリストを作成する
                        (do ((aaa nil)
                            (action action (cdr action)))
                            ((null action) aaa)
                            (setq aaa (append aaa (newMatchAux action))))))
                      (defun newMatchAux (action)
                        ;確信度の計算
                        (cond ((null (cdr action))
                              (cond ((numberp (car action)) (list (* (car action) matchRatio)))
                                    (t (append (list (car action)) (list matchRatio)))))
                              (t (list (car action)))))
                      (defun remember (p x y z)
                        ;確信度の大きい順に格納する
                        (cond ((< z 0.01)
                              p)
                              ((null p)
                              (cons (cons x (cons y (cons (r100 z) nil))) p))
                              ((equal y (cadr p))
                              (cond ((> z (caddr p))
                                    (rplaca (car p) x)
                                    (rplacd (cadr p) (cons (r100 z) nil))
                                    p)
                                    (t p)))
                              ((> z (caddr p))
                              (push (cons x (cons y (cons (r100 z) nil))) p)
                              (del_list y p)
                              p)
                              (t (cons (car p) (remember (cdr p) x y z))))))
                      (defun del_list (y p)
                        ;リストからの削除
                        (cond ((null (cdr p)) nil)
                              ((equal (caddr p) y) (rplacd p (caddr p)))
                              (t (del_list y (cdr p))))))
                      ;---( 利用したルールのリストを生成する )-----
                      (defun feed_to_rules_used (rule_name if_list then_list a_list_stream)

```







```

**** mulisp86 version ***** 92.11.14 *****
;
; File name : TOOL. LSP
; Feature   : リスプ・ユーティリティ
;
*****

; ストリームの処理

(defun combine_streams (s1 s2) (append s1 s2))

(defun add_to_stream (e s) (cons e s))

(defun first_of_stream (s) (car s))

(defun rest_of_stream (s) (cdr s))

(defun empty_stream_p (s) (null s))

(defun make_empty_stream () nil)

(defun read_file (file_name)
; ファイルの読みだし
  ((rds file_name)
   (setq text nil)
   (loop (setq text1 (read))
         ((equal text1 '(rds)) (rds))
         (setq text (cons text text1)))
   text))

(defun print_list (list)
; リスト中のリストを1行毎に表示する
  (cond ((null list) t)
        ((prtsen (car list))
         (print_list (cdr list)))))

(defun print_char (message)
; リスト中のリストを括弧をはずして1行毎に表示する
  (cond ((null message) t)
        ((prtsen (car message))
         (print_char (cdr message)))))

(defun prtsen (message)
; リスト中のアトムを括弧をはずして表示する
  (cond ((null message) (terpri) t)
        ((prnl (car message))
         (spaces 1)
         (prtsen (cdr message)))))

(defun prtsen1 (message)
  (cond ((null message) t)
        ((prnl (car message))
         (spaces 1)
         (prtsen1 (cdr message)))))

(defun print_question (message)
  (print_char message)
  (prnl '-->))

(defun print_y_n_question (message)
  (print_char (list (append (car message) '(?) '(<yes/no>))))
  (prnl '-->))

(defun Y-OR-N-P-2 (MSG
  CHAR READ-CHAR RDS WRS )
  ( ( (NULL MSG))
    (FRESH-LINE)
    (WRITE-STRING (PACK* MSG) ) )
  (CLEAR-INPUT)
  (LOOP
   (SETQ CHAR (CHAR-UPCASE (READ-CHAR)))
   ((EQ CHAR 'Y) (WRITE-LINE CHAR) T)
   ((EQ CHAR 'N) (WRITE-LINE CHAR) NIL)
   (WRITE-BYTE 7) ) )

(rds b:rule)

```

```

**** mulisp86 version ***** 92.11.14 *****
;
; File name : NDI_M. KDB
; Feature   : マスター・フレーム
;
*****

(setq *masterframes* nil)
(setq *frames* nil)

(store_frames '(
  (weld_condition ; 溶接条件の入力と推論を制御するフレーム
    (ako (value master))
    (begin (value subclass))
    (subclasses (value (get_w_data
                        get_ut_data
                        weld_rules1
                        weld_rules2))))
  (ut_test ; 超音波探傷試験の結果の入力と推論を制御するフレーム
    (ako (value master))
    (begin (value subclass))
    (subclasses (value (choice_groove
                        ut_rules))))
  (estimation_flaw ; 欠陥の等級判定と欠陥の寸法の推定を制御するフレーム
    (ako (value master))
    (begin (value subclass
              estimation
              display_e
              msg_e))
    (subclasses (value (figure_rules
                        flaw_class))
    (data_name (value (溶接方法 材料 継手形状 開先形状 溶接層
                        裏当金 裏はつり
                        板厚 開先角 表開先角 裏開先角
                        R間隔 開先深 表開先深 裏開先深
                        屈折角 周波数 探傷面 探触子
                        Y B 路程
                        E 高さ (H線基準) 指示長さ 前後走査 振子走査
                        両側探傷 エコー 特性曲線
                        表面欠陥 欠陥位置
                        Z D))
    (result_name (value 欠陥 欠陥の等級))))
  (choice_groove ; 開先形状に対応するデータ変換フレームを選択する
    (ako (value master))
    (begin (value choice_g))
    (V形開先 (value data_transform_sv))
    (X形開先 (value data_transform_dv))
    (レ形開先 (value data_transform_sb))
    (K形開先 (value data_transform_db))))))

(rds b:ndi_g.kdb)

```



```

**** mulisp86 version ***** 92.11.14 *****
:
: File name : NDI G. KDB
: Feature : 溶接条件および超音波探傷に関するデータの獲得
:
: *****
(store_frames

  ((get_data ;データ獲得のマスターフレーム
    (ako (value master))
    (begin (value get_d))
    (result (value))
    (identify (value 入力するデータの名称))
    (order_list (value 入力するデータの順番を表すリスト))
    (get_ndata_list (value 言語データの入力項目を示すリスト))
    (get_vdata_list (value 数値データの入力項目を示すリスト))
    (judg_list (value 入力値に基づく質問項目の選択を示すリスト))
    (subclasses (value get_w_data)))

  (get_w_data ;溶接条件に関するデータ獲得のフレーム
    (ako (value get_data))
    (result1 (value *input_w1*))
    (result2 (value *input_w2*))
    (identify (value 溶接に関するデータを質問文にしたがって入力してください))
    (order_list (value 溶接方法 材料 継手形状 開先形状))
    (get_ndata_list (value 溶接方法 材料 継手形状 開先形状 裏当金
      溶接層 裏はつり))
    (get_vdata_list (value 板厚 R間隔
      開先角 表開先角 裏開先角
      開先深 表開先深 裏開先深))

    (judg_list (value (開先形状 (V形開先 groove_s)
      (X形開先 groove_d)
      (レ形開先 groove_s)
      (K形開先 groove_d))
      (裏当金 (ない atenal))) )
    (result2_data (value 板厚 R間隔
      開先角 表開先角 裏開先角
      開先深 表開先深 裏開先深))

    (溶接方法 (value (溶接方法を選択してください)
      (1 被覆アーク溶接)
      (2 半自動炭酸ガスアーク溶接)))
    (材料 (value (材料を選択してください)
      (1 SS鋼)
      (2 SM鋼)))
    (開先形状 (value (開先形状を選択してください)
      (1 V形開先)
      (2 レ形開先)
      (3 X形開先)
      (4 K形開先)))
    (継手形状 (value (継手形状を選択してください)
      (1 突合せ継手)
      (2 かど継手)
      (3 T字継手)
      (4 +字継手)))

    (groove_s (value 溶接層 裏当金 板厚 開先角 R間隔 開先深))
    (groove_d (value 溶接層 板厚 表開先角 裏開先角 R間隔
      表開先深 裏開先深))

    (溶接層 (value (溶接層は多層ですか)
      (1 多層)
      (2 単層)))
    (裏当金 (value (裏当金がありますか)
      (1 ある)
      (2 ない)))

    (atenal (value 裏はつり))

    (裏はつり (value (裏はつりをしますか)
      (1 する)
      (2 しない)))

    (板厚 (value 板厚を入力して下さい <mm>))
    (開先角 (value 開先角を入力して下さい° ))
    (表開先角 (value 表面開先角を入力して下さい° ))
    (裏開先角 (value 裏面開先角を入力して下さい° ))
    (R間隔 (value ルート間隔を入力して下さい <mm>))
    (開先深 (value 開先深さを入力して下さい <mm>))
    (表開先深 (value 表面開先深さを入力して下さい <mm>))
    (裏開先深 (value 裏面開先深さを入力して下さい <mm>))

  (get_data2 ;データ獲得のマスターフレーム
    (ako (value master))

```

```

    (begin (value get_d2))
    (result (value データの出力先))
    (identify (value 入力するデータの名称))
    (order_list (value 入力するデータの順番を表すリスト))
    (get_ndata_list (value 言語データの入力項目を示すリスト))
    (get_vdata_list (value 数値データの入力項目を示すリスト))
    (judg_list (value 入力値に基づく質問項目の選択を示すリスト))
    (subclasses (value get_ut_data)))

  (get_ut_data ;超音波探傷に関するデータ獲得のフレーム
    (ako (value get_data2))
    (result1 (value *input_ut1*))
    (result2 (value *input_ut2*))
    (identify (value 超音波探傷のデータを質問文にしたがって入力してください))
    (order_list (value 屈折角 周波数
      探傷面 探触子
      Y B路程
      E高さ (H線基準) 指示長さ 前後走査 振子走査
      両側探傷 エコー 特性曲線 表面欠陥))
    (get_vdata_list (value 屈折角 周波数 Y B路程 E高さ (H線基準) 指示長さ))
    (get_ndata_list (value 探傷面 探触子 前後走査 振子走査 両側探傷
      エコー 特性曲線 表面欠陥 欠陥位置))

    (judg_list (value (表面欠陥 (ある face_flaw)))
    (result2_data (value 屈折角 周波数 探傷面 探触子 Y B路程
      E高さ (H線基準) 指示長さ 前後走査))

    (屈折角 (value 探触子の屈折角を入力してください))
    (周波数 (value 超音波の周波数を入力してください <MHz>))
    (Y (value 探触子・溶接部距離 (Y)を入力してください <mm>))
    (B路程 (value ビーム路程 (W)を入力してください <mm>))
    (E高さ (H線基準) (value エコー高さをH線からの差で入力してください))
    (指示長さ (value 欠陥指示長さを入力してください <mm>))

    (face_flaw (value 欠陥位置))

    (探傷面 (value (探傷面を以下から選択してください)
      (1 表面)
      (2 裏面)))
    (探触子 (value (探触子の位置を以下から選択してください)
      (1 開先を施した材の上)
      (2 開先を施していない材の上)))
    (前後走査 (value (前後走査における6dBダウソ欠陥指示長さを以下から選択してください)
      (1 10mm未満)
      (2 10mm以上20mm以下)
      (3 20mmを超える)))
    (振子走査 (value (振子走査 (振子角15度)におけるエコー低下を以下から選択してください。)
      (1 8dB未満)
      (2 8dB以上15dB以下)
      (3 15dBを超える)))
    (両側探傷 (value (両側探傷したときのエコー高さを以下から選択してください)
      (1 10dBより明らかに小さい)
      (2 10dBより明らかに大きい)
      (3 10dB前後である)
      (4 測定できないまたは測定していない)))
    (エコー (value (エコーの波形を以下から選択してください)
      (1 明らかにギザギザでない)
      (2 ギザギザかどうかはつきりしない)
      (3 明らかにギザギザである)
      (4 いろいろなビーム路程で立ち上がるかギザギザであるかはつきりしない)
      (5 明らかにいろいろなビーム路程で立ち上がる)))
    (特性曲線 (value (距離振幅特性曲線はA2-φ4より緩やかですか)
      (1 明らかにA2-φ4より緩やかである)
      (2 A2-φ4とほぼ同じ傾きかもしくは急である)
      (3 A2-φ4より緩やかかどうかはつきりしない)
      (4 測定していない)))
    (表面欠陥 (value (溶接部表面に欠陥はありますか)
      (1 ある)
      (2 ない)))
    (欠陥位置 (value (表面の欠陥位置はどこにありますか)
      (1 止端である)
      (2 止端でない)))

  (get_data_unit ;
    (name (value 板厚 開先角 表開先角 裏開先角
      R間隔 開先深 表開先深 裏開先深
      屈折角 周波数
      Y B路程
      E高さ (H線基準) 指示長さ
      Z D)))

  (板厚
    (ako (value get_data_unit))
    (unit (value mm)))

  (開先角
    (ako (value get_data_unit))
    (unit (value °)))

  (表開先角

```



```

(ako (value get_data_unit))
(unit (value °)))

(裏開先角
(ako (value get_data_unit))
(unit (value °)))

(R 間隔
(ako (value get_data_unit))
(unit (value mm)))

(開先深
(ako (value get_data_unit))
(unit (value mm)))

(表開先深
(ako (value get_data_unit))
(unit (value mm)))

(裏開先深
(ako (value get_data_unit))
(unit (value mm)))

(屈折角
(ako (value get_data_unit))
(unit (value °)))

(周波数
(ako (value get_data_unit))
(unit (value MHz)))

(Y
(ako (value get_data_unit))
(unit (value mm)))

(B 路程
(ako (value get_data_unit))
(unit (value mm)))

(E 高さ (H 線基準)
(ako (value get_data_unit))
(unit (value dB)))

(指示長さ
(ako (value get_data_unit))
(unit (value mm)))

(Z
(ako (value get_data_unit))
(unit (value mm)))

(D
(ako (value get_data_unit))
(unit (value mm)))

(data_transform ;データ変換用フレーム
(ako (value master))
(begin (value data_t))
(groove_type (value type of groove)))

(data_transform_sv
(ako (value data_transform))
(groove_type (value single_v)))

(data_transform_dv
(ako (value data_transform))
(groove_type (value double_v)))

(data_transform_sb
(ako (value data_transform))
(groove_type (value single_b)))

(data_transform_db
(ako (value data_transform))
(groove_type (value double_b))))

(rds b:ndi_w.kdb)

```

```

**** mulisp86 version **** 92.11.14 ****
;
; File name : NDI W. KDB
; Feature : 溶接条件より欠陥の種類と位置を推定するためのルール
;
*****

(setq metarule '(weld_condition
  ut_test
  estimation_flaw))

(store_frames
  '(
    (weld_rules1 ;溶接条件に関するプロダクションルールを格納するフレーム
      (ako (value trigrules))
      (forward (value *input_wl*))
      (input (value *input_wl*))
      (result (value *result_wl*))
      (whylog (value *whylog_wl*))
      (rules (value
        (rule S S 鋼 V 形密集ブローホール
          (if ((開先形状は V 形開先))
              ((裏当金はない))
              ((材料は S S 鋼)))
          (then ((開先による欠陥は密集ブローホールである
                開先による欠陥位置は溶接金属部である))))
        (rule S S 鋼 V 形単独ブローホール
          (if ((開先形状は V 形開先))
              ((裏当金はない))
              ((材料は S S 鋼)))
          (then ((開先による欠陥は単独ブローホールである
                開先による欠陥位置は溶接金属部である))))
        (rule S S 鋼 V 形開先面スラグ巻き込み 1
          (if ((開先形状は V 形開先))
              ((裏当金はない))
              ((溶接層は単層))
              ((材料は S S 鋼)))
          (then ((開先による欠陥は開先面スラグ巻き込みである
                開先による欠陥位置は開先部である))))
        (rule S S 鋼 V 形開先面スラグ巻き込み 2
          (if ((開先形状は V 形開先))
              ((裏当金はない))
              ((材料は S S 鋼))
              ((溶接層は多層)))
          (then ((開先による欠陥は開先面スラグ巻き込みである
                開先による欠陥位置は開先部である))))
        (rule S S 鋼 V 形スラグ巻き込み
          (if ((開先形状は V 形開先))
              ((裏当金はない))
              ((材料は S S 鋼))
              ((溶接層は単層)))
          (then ((開先による欠陥はスラグ巻き込みである
                開先による欠陥位置は溶接金属部である))))
        (rule S S 鋼 V 形層間スラグ巻き込み
          (if ((開先形状は V 形開先))
              ((裏当金はない))
              ((材料は S S 鋼))
              ((溶接層は多層)))
          (then ((開先による欠陥は層間スラグ巻き込みである
                開先による欠陥位置は溶接金属部である))))
        (rule S S 鋼 V 形開先面融合不良 1
          (if ((開先形状は V 形開先))
              ((裏当金はない))
              ((材料は S S 鋼))
              ((溶接層は単層)))
          (then ((開先による欠陥は開先面融合不良である
                開先による欠陥位置は開先部である))))
        (rule S S 鋼 V 形開先面融合不良 2
          (if ((開先形状は V 形開先))
              ((裏当金はない))
              ((材料は S S 鋼))
              ((溶接層は多層)))
          (then ((開先による欠陥は開先面融合不良である
                開先による欠陥位置は開先部である))))
        (rule S S 鋼 V 形層間融合不良
          (if ((開先形状は V 形開先))
              ((裏当金はない))
              ((材料は S S 鋼))
              ((溶接層は多層)))
          (then ((開先による欠陥は層間融合不良である
                開先による欠陥位置は溶接金属部である))))
        (rule S S 鋼 V 形ルート溶け込み不良
          (if ((開先形状は V 形開先))
              ((裏当金はない))
              ((材料は S S 鋼)))
          (then ((開先による欠陥はルート溶け込み不良である
                開先による欠陥位置はルート部である))))
        (rule S S 鋼 V 形ルート割れ

```



[illegible][illegible]



[illegible]



(  
((真当金はな))  
((材接層はSM鋼)))  
(then ((開先による欠陥位置は層間スラグ巻き込みである  
(rule S M鋼先V形はあ閉合不良1  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))  
(then ((開先による欠陥位置は開先面融合不良である  
(rule S M鋼先V形はあ閉合不良2  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))  
(then ((開先による欠陥位置は開先面融合不良である  
(rule S M鋼先V形はあ閉合不良  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))  
(then ((開先による欠陥位置は層間融合金属部である  
(rule S M鋼先V形はあ閉合不良  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))  
(then ((開先による欠陥位置はルート溶け込み不良である  
(rule S M鋼先V形はあ閉合不良  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))  
(then ((開先による欠陥位置はルート割れである  
(rule S M鋼先V形はあ閉合不良  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))  
(then ((開先による欠陥位置はビード下割れである  
(rule S M鋼先V形はあ閉合不良  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))  
(then ((開先による欠陥位置はH A Z部である  
(rule S M鋼先V形はあ閉合不良  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))  
(then ((開先による欠陥位置は内部集積ブローホールである  
(rule S M鋼先V形はあ閉合不良  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))  
(then ((開先による欠陥位置は密集ブローホールである  
(rule S M鋼先V形はあ閉合不良  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))  
(then ((開先による欠陥位置は単独ブローホールである  
(rule S M鋼先V形はあ閉合不良  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))  
(then ((開先による欠陥位置は開先面スラグ巻き込みである  
(rule S M鋼先V形はあ閉合不良  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))  
(then ((開先による欠陥位置は開先面スラグ巻き込みである  
(rule S M鋼先V形はあ閉合不良  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))  
(then ((開先による欠陥位置は開先面スラグ巻き込みである  
(rule S M鋼先V形はあ閉合不良  
(if ((開先V形状はV形開先)  
((裏当金はな))  
((材料はSM鋼)))  
(溶接層は多層)))







```

((if ((開先形状はレ形開先))
  ((裏当金は無い))
  ((材料はSM鋼)))
  (then ((開先による欠陥はルート溶け込み不良である
(rule SM鋼先形による欠陥位置はルート部である)))
  (if ((開先形状はレ形開先))
    ((裏当金は無い))
    ((材料はSM鋼)))
    (then ((開先による欠陥はルート割れである
(rule SM鋼先形による欠陥位置はルート部である)))
    (if ((開先形状はレ形開先))
      ((裏当金は無い))
      ((材料はSM鋼)))
      (then ((開先による欠陥位置はビード下割れである
(rule SM鋼先形による欠陥位置はHAZ部である)))
      (if ((開先形状はレ形開先))
        ((裏当金は無い))
        ((材料はSM鋼)))
        (then ((開先による欠陥位置は内部割れである
(rule SM鋼先形による欠陥位置は溶接金属部である)))
        (if ((開先形状はレ形開先))
          ((裏当金は無い))
          ((材料はSM鋼)))
          (then ((開先による欠陥位置は密集ブローホールである
(rule SM鋼先形による欠陥位置はブローホール
          (if ((開先形状はレ形開先))
            ((裏当金は無い))
            ((材料はSM鋼)))
            (then ((開先による欠陥位置は単独ブローホールである
(rule SM鋼先形による欠陥位置は溶接金属部である)))
            (if ((開先形状はレ形開先))
              ((裏当金は無い))
              ((材料はSM鋼)))
              (then ((開先による欠陥位置は単独ブローホールである
(rule SM鋼先形による欠陥位置は溶接金属部である)))
              (if ((開先形状はレ形開先))
                ((裏当金は無い))
                ((材料はSM鋼)))
                (then ((開先による欠陥位置は開先面スラグ巻き込みである
(rule SM鋼先形による欠陥位置は開先面スラグ巻き込み2
                (if ((開先形状はレ形開先))
                  ((裏当金は無い))
                  ((材料はSM鋼)))
                  (then ((開先による欠陥位置は開先面スラグ巻き込みである
(rule SM鋼先形による欠陥位置は開先面スラグ巻き込み
                  (if ((開先形状はレ形開先))
                    ((裏当金は無い))
                    ((材料はSM鋼)))
                    (then ((開先による欠陥位置はスラグ巻き込みである
(rule SM鋼先形による欠陥位置はスラグ巻き込み
                    (if ((開先形状はレ形開先))
                      ((裏当金は無い))
                      ((材料はSM鋼)))
                      (then ((開先による欠陥位置は層間スラグ巻き込みである
(rule SM鋼先形による欠陥位置は層間スラグ巻き込み1
                      (if ((開先形状はレ形開先))
                        ((裏当金は無い))
                        ((材料はSM鋼)))
                        (then ((開先による欠陥位置は開先面融合不良1
(rule SM鋼先形による欠陥位置は開先面融合不良2
                        (if ((開先形状はレ形開先))
                          ((裏当金は無い))
                          ((材料はSM鋼)))
                          (then ((開先による欠陥位置は開先面融合不良である
(rule SM鋼先形による欠陥位置は開先面融合不良
                          (if ((開先形状はレ形開先))
                            ((裏当金は無い))
                            ((材料はSM鋼)))
                            (then ((開先による欠陥位置は層間融合不良である
(rule SM鋼先形による欠陥位置は層間融合不良
                            (if ((開先形状はレ形開先))
                              ((裏当金は無い))
                              ((材料はSM鋼)))
                              (then ((開先による欠陥位置はルート溶け込み不良
(rule SM鋼先形による欠陥位置はルート溶け込み不良

```

[illegible]



```

(if ((溶接方法は被覆アーク溶接))
(then ((溶接方法による欠陥はルート溶け込み不良である)))
(rule 被覆アーク溶接ルート割れ
(if ((溶接方法は被覆アーク溶接))
(then ((溶接方法による欠陥はルート割れである)))
(rule 被覆アーク溶接ビード下割れ
(if ((溶接方法は被覆アーク溶接))
(then ((溶接方法による欠陥はビード下割れである)))
(rule 被覆アーク溶接内部割れ
(if ((溶接方法は被覆アーク溶接))
(then ((溶接方法による欠陥は内部割れである)))
(rule 被覆アーク溶接単独ブローホール
(if ((溶接方法は被覆アーク溶接))
(then ((溶接方法による欠陥は単独ブローホールである)))
(rule 被覆アーク溶接密集ブローホール
(if ((溶接方法は被覆アーク溶接))
(then ((溶接方法による欠陥は密集ブローホールである)))
(rule 被覆アーク溶接開先面スラグ巻き込み
(if ((溶接方法は被覆アーク溶接))
(then ((溶接方法による欠陥は開先面スラグ巻き込みである)))
(rule 被覆アーク溶接層間スラグ巻き込み
(if ((溶接方法は被覆アーク溶接))
(then ((溶接方法による欠陥は層間スラグ巻き込みである)))
(rule 被覆アーク溶接スラグ巻き込み
(if ((溶接方法は被覆アーク溶接))
(then ((溶接方法による欠陥はスラグ巻き込みである)))
(rule 被覆アーク溶接内部溶け込み不良
(if ((溶接方法は被覆アーク溶接))
(then ((溶接方法による欠陥は内部溶け込み不良である)))
(rule 被覆アーク溶接開先面融合不良
(if ((溶接方法は被覆アーク溶接))
(then ((溶接方法による欠陥は開先面融合不良である)))
(rule 被覆アーク溶接層間融合不良
(if ((溶接方法は被覆アーク溶接))
(then ((溶接方法による欠陥は層間融合不良である)))

(rule 半自動炭酸ガスアーク溶接ルート溶け込み不良
(if ((溶接方法は半自動炭酸ガスアーク溶接))
(then ((溶接方法による欠陥はルート溶け込み不良である)))
(rule 半自動炭酸ガスアーク溶接ルート割れ
(if ((溶接方法は半自動炭酸ガスアーク溶接))
(then ((溶接方法による欠陥はルート割れである)))
(rule 半自動炭酸ガスアーク溶接ビード下割れ
(if ((溶接方法は半自動炭酸ガスアーク溶接))
(then ((溶接方法による欠陥はビード下割れである)))
(rule 半自動炭酸ガスアーク溶接内部割れ
(if ((溶接方法は半自動炭酸ガスアーク溶接))
(then ((溶接方法による欠陥は内部割れである)))
(rule 半自動炭酸ガスアーク溶接密集ブローホール
(if ((溶接方法は半自動炭酸ガスアーク溶接))
(then ((溶接方法による欠陥は密集ブローホールである)))
(rule 半自動炭酸ガスアーク溶接単独ブローホール
(if ((溶接方法は半自動炭酸ガスアーク溶接))
(then ((溶接方法による欠陥は単独ブローホールである)))
(rule 半自動炭酸ガスアーク溶接開先面融合不良
(if ((溶接方法は半自動炭酸ガスアーク溶接))
(then ((溶接方法による欠陥は開先面融合不良である)))
(rule 半自動炭酸ガスアーク溶接層間融合不良
(if ((溶接方法は半自動炭酸ガスアーク溶接))
(then ((溶接方法による欠陥は層間融合不良である)))
(rule 半自動炭酸ガスアーク溶接内部溶け込み不良
(if ((溶接方法は半自動炭酸ガスアーク溶接))
(then ((溶接方法による欠陥は内部溶け込み不良である)))
(rule 半自動炭酸ガスアーク溶接開先面スラグ巻き込み
(if ((溶接方法は半自動炭酸ガスアーク溶接))
(then ((溶接方法による欠陥は開先面スラグ巻き込みである)))
(rule 半自動炭酸ガスアーク溶接層間スラグ巻き込み
(if ((溶接方法は半自動炭酸ガスアーク溶接))
(then ((溶接方法による欠陥は層間スラグ巻き込みである)))
(rule 半自動炭酸ガスアーク溶接スラグ巻き込み
(if ((溶接方法は半自動炭酸ガスアーク溶接))
(then ((溶接方法による欠陥はスラグ巻き込みである))))))

```

```

(store_frames
'

```

```

(weld_rules2 ;溶接条件に関するプロダクションルールを格納するフレーム
(ako (value trigrules2))
(forward (value weld_rules1))
(input (value *result_w1*))
(result (value *result_w2*))
(whylog (value *whylog_w2*))
(rules (value
(rule 溶接条件ルート溶け込み不良
(if ((開先による欠陥はルート溶け込み不良である
開先による欠陥位置はルート部である))
((溶接方法による欠陥はルート溶け込み不良である)))
(then ((欠陥はルート溶け込み不良である)
(欠陥位置はルート部である))))))
(rule 溶接条件ルート割れ

```

```

(if ((開先による欠陥はルート割れである
開先による欠陥位置はルート部である))
((溶接方法による欠陥はルート割れである)))
(then ((欠陥はルート割れである)
(欠陥位置はルート部である))))))
(rule 溶接条件密集ブローホール
(if ((開先による欠陥は密集ブローホールである
開先による欠陥位置は溶接金属部である))
((溶接方法による欠陥は密集ブローホールである)))
(then ((欠陥は密集ブローホールである)
(欠陥位置は溶接金属部である))))))
(rule 溶接条件単独ブローホール
(if ((開先による欠陥は単独ブローホールである
開先による欠陥位置は溶接金属部である))
((溶接方法による欠陥は単独ブローホールである)))
(then ((欠陥は単独ブローホールである)
(欠陥位置は溶接金属部である))))))
(rule 溶接条件開先面スラグ巻き込み
(if ((開先による欠陥は開先面スラグ巻き込みである
開先による欠陥位置は開先部である))
((溶接方法による欠陥は開先面スラグ巻き込みである)))
(then ((欠陥は開先面スラグ巻き込みである)
(欠陥位置は開先部である))))))
(rule 溶接条件層間スラグ巻き込み
(if ((開先による欠陥は層間スラグ巻き込みである
開先による欠陥位置は溶接金属部である))
((溶接方法による欠陥は層間スラグ巻き込みである)))
(then ((欠陥は層間スラグ巻き込みである)
(欠陥位置は溶接金属部である))))))
(rule 溶接条件スラグ巻き込み
(if ((開先による欠陥はスラグ巻き込みである
開先による欠陥位置は溶接金属部である))
((溶接方法による欠陥はスラグ巻き込みである)))
(then ((欠陥はスラグ巻き込みである)
(欠陥位置は溶接金属部である))))))
(rule 溶接条件内部溶け込み不良
(if ((開先による欠陥は内部溶け込み不良である
開先による欠陥位置は溶接金属部である))
((溶接方法による欠陥は内部溶け込み不良である)))
(then ((欠陥は内部溶け込み不良である)
(欠陥位置は溶接金属部である))))))
(rule 溶接条件ビード下割れ
(if ((開先による欠陥はビード下割れである
開先による欠陥位置はHAZ部である))
((溶接方法による欠陥はビード下割れである)))
(then ((欠陥はビード下割れである)
(欠陥位置はHAZ部である))))))
(rule 溶接条件内部割れ
(if ((開先による欠陥は内部割れである
開先による欠陥位置は溶接金属部である))
((溶接方法による欠陥は内部割れである)))
(then ((欠陥は内部割れである)
(欠陥位置は溶接金属部である))))))
(rule 溶接条件開先面融合不良
(if ((開先による欠陥は開先面融合不良である
開先による欠陥位置は開先部である))
((溶接方法による欠陥は開先面融合不良である)))
(then ((欠陥は開先面融合不良である)
(欠陥位置は開先部である))))))
(rule 溶接条件層間融合不良
(if ((開先による欠陥は層間融合不良である
開先による欠陥位置は溶接金属部である))
((溶接方法による欠陥は層間融合不良である)))
(then ((欠陥は層間融合不良である)
(欠陥位置は溶接金属部である))))))

```

```

(rds b:ndi_ut.kdb)

```



\*\*\*\* mulisp86 version \*\*\*\* 92.11.14 \*\*\*\*

File name : NDI UT. KDB  
Feature : 超音波探傷に関するデータより欠陥種類を推定するルール

```
(store_frames
  (
    (trigrules ;プロダクションルールを使用して推論を実行するフレーム
      (ako (value master))
      (begin (value t_rules))
      (forward (value))
      (input (value))
      (result (value 結果を格納する変数名))
      (whylog (value 利用したルールを格納する変数名))
      (rules (value (rule ルール名
        (if (仮定))
        (then (結論))))))
    (trigrules2
      (ako (value master))
      (begin (value t_rules2)))
    (ut_rules ;超音波探傷に関するプロダクションルールを格納したフレーム
      (ako (value trigrules))
      (forward (value *input_w1* *result_ut1* *result_ut3* *input_ut1*
        weld_rules2))
      (input (value *input_w1* *result_w2* *input_ut1* *result_ut3*
        *result_ut1*))
      (result (value *result_ut*))
      (whylog (value *whylog_ut*))
      (rules (value
        (rule 密集ブローホール1
          (if ((欠陥位置は ルート部である))
            ((欠陥位置は 密集ブローホールである)
             (欠陥位置は 溶接金属部である))
            ((両側探傷は 10dB以下である))
            ((エコーは いろいろなビーム路程で立ち上がる)))
          (then ((欠陥は 密集ブローホールである))))
          (rule 密集ブローホール2
            (if ((欠陥位置は ルート部である))
              ((欠陥位置は 密集ブローホールである)
               (欠陥位置は 溶接金属部である))
              ((両側探傷は 10dB以下である))
              ((振子走査は 小さい))
              ((エコーは ギザギザである))
              ((点数は 約6以下である)))
            (then ((欠陥は 密集ブローホールである) 0.6)))
          (rule 単独ブローホール1
            (if ((欠陥位置は ルート部である))
              ((欠陥位置は 単独ブローホールである)
               (欠陥位置は 溶接金属部である))
              ((両側探傷は 10dB以下である))
              ((振子走査は 小さい))
              ((エコーは ギザギザでない))
              ((点数は 約6以下である)))
            (then ((欠陥は 単独ブローホールである))))
          (rule ルート溶け込み不良
            (if ((欠陥位置は ルート部である))
              ((欠陥位置は ルート溶け込み不良である)
               (欠陥位置は ルート部である))
              ((両側探傷は 10dB以下である))
              ((点数は 約10以上である)))
            (then ((欠陥は ルート溶け込み不良である))))
          (rule ルート割れ
            (if ((欠陥位置は ルート部である))
              ((欠陥位置は ルート割れである)
               (欠陥位置は ルート部である))
              ((両側探傷は 10dB以下である))
              ((点数は 約10以上である)))
            (then ((欠陥は ルート割れである) 0.3)))
          (rule 密集ブローホール3
            (if ((欠陥位置は 溶接金属部である))
              ((欠陥位置は 密集ブローホールである)
               (欠陥位置は 溶接金属部である))
              ((両側探傷は 10dB以下である))
              ((エコーは いろいろなビーム路程で立ち上がる)))
            (then ((欠陥は 密集ブローホールである))))
          (rule 密集ブローホール4
            (if ((欠陥位置は 溶接金属部である))
              ((欠陥位置は 密集ブローホールである)
               (欠陥位置は 溶接金属部である))
              ((両側探傷は 10dB以下である))
              ((振子走査は 小さい))
              ((エコーは ギザギザである))
              ((点数は 約6以下である)))
            (then ((欠陥は 密集ブローホールである))))
        )
      )
    )
  )
```

```
(then ((欠陥は 密集ブローホールである) 0.6)))
(rule 単独ブローホール2
  (if ((欠陥位置は 溶接金属部である))
    ((欠陥位置は 単独ブローホールである)
     (欠陥位置は 溶接金属部である))
    ((両側探傷は 10dB以下である))
    ((振子走査は 小さい))
    ((エコーは ギザギザでない))
    ((点数は 約6以下である)))
  (then ((欠陥は 単独ブローホールである))))
(rule 層間スラグ巻き込み
  (if ((欠陥位置は 溶接金属部である))
    ((欠陥位置は 層間スラグ巻き込みである)
     (欠陥位置は 溶接金属部である))
    ((点数は 約7以上約9以下である))
    ((エコーは ギザギザに近い))
    (then ((欠陥は 層間スラグ巻き込みである))))
(rule スラグ巻き込み
  (if ((欠陥位置は 溶接金属部である))
    ((欠陥位置は スラグ巻き込みである)
     (欠陥位置は 溶接金属部である))
    ((点数は 約7以上約9以下である))
    ((エコーは ギザギザに近い))
    (then ((欠陥は スラグ巻き込みである))))
(rule 層間融合不良
  (if ((欠陥位置は 溶接金属部である))
    ((欠陥位置は 層間融合不良である)
     (欠陥位置は 溶接金属部である))
    ((エコーは ギザギザでないに近い))
    ((点数は 約7以上約9以下である))
    (then ((欠陥は 層間融合不良である))))
(rule 内部溶け込み不良
  (if ((欠陥位置は 溶接金属ルート部である))
    ((欠陥位置は 内部溶け込み不良である)
     (欠陥位置は 溶接金属部である))
    ((両側探傷は 10dB以下である))
    ((エコーは いろいろなビーム路程で立ち上がらない))
    ((点数は 約10以上である))
    ((特性曲線は A2-φ4より緩やかである))
    (then ((欠陥は 内部溶け込み不良である))))
(rule 内部割れ
  (if ((欠陥位置は 溶接金属部である))
    ((欠陥位置は 内部割れである)
     (欠陥位置は 溶接金属部である))
    ((エコーは いろいろなビーム路程で立ち上がらない))
    ((点数は 約10以上である))
    ((特性曲線は A2-φ4より緩やかでない))
    (then ((欠陥は 内部割れである) 0.3)))
(rule 密集ブローホール5
  (if ((欠陥位置は 開先部である))
    ((欠陥位置は 密集ブローホールである)
     (欠陥位置は 溶接金属部である))
    ((両側探傷は 10dB以下である))
    ((エコーは いろいろなビーム路程で立ち上がる)))
  (then ((欠陥は 密集ブローホールである))))
(rule 密集ブローホール6
  (if ((欠陥位置は 開先部である))
    ((欠陥位置は 密集ブローホールである)
     (欠陥位置は 溶接金属部である))
    ((両側探傷は 10dB以下である))
    ((振子走査は 小さい))
    ((エコーは ギザギザである))
    ((点数は 約6以下である))
    (then ((欠陥は 密集ブローホールである) 0.6)))
(rule 単独ブローホール3
  (if ((欠陥位置は 開先部である))
    ((欠陥位置は 単独ブローホールである)
     (欠陥位置は 溶接金属部である))
    ((両側探傷は 10dB以下である))
    ((振子走査は 小さい))
    ((エコーは ギザギザでない))
    ((点数は 約6以下である)))
  (then ((欠陥は 単独ブローホールである))))
(rule 開先面スラグ巻き込み
  (if ((欠陥位置は 開先部である))
    ((欠陥位置は 開先面スラグ巻き込みである)
     (欠陥位置は 開先部である))
    ((エコーは ギザギザに近い))
    ((点数は 約7以上約9以下である))
    (then ((欠陥は 開先面スラグ巻き込みである))))
(rule 開先面融合不良
  (if ((欠陥位置は 開先部である))
    ((欠陥位置は 開先面融合不良である)
     (欠陥位置は 開先部である))
    ((エコーは ギザギザでないに近い))
    ((点数は 約7以上約9以下である))
    (then ((欠陥は 開先面融合不良である))))
(rule ビード下割れ
```



```

((if ((欠陥はビード下割れである)
      (欠陥位置はHAZ部である)))
  ((欠陥位置はHAZ部である)))
  (then ((欠陥はビード下割れである) 0.3)))
(rule 止端割れ
  (if ((欠陥位置は止端部である)
        (表面欠陥はある)))
    (then ((欠陥は止端割れである))))
(rule アンダーカット
  (if ((欠陥位置は止端部である)
        (表面欠陥はある)))
    (then ((欠陥はアンダーカットである))))
(rule ビット
  (if ((欠陥位置は止端部でない)
        (表面欠陥はある)))
    (then ((欠陥はビットである))))
(rule ビード表面割れ
  (if ((欠陥位置は止端部でない)
        (表面欠陥はある)))
    (then ((欠陥はビード表面割れである))))

(rule 妨害エコー1
  (if ((開先形状はV形開先))
        ((両側探傷は10dBより大きい))
        ((欠陥位置は裏波付近である)))
    (then ((欠陥ではなく妨害エコー1である))) )
(rule 妨害エコー2
  (if ((開先形状はV形開先))
        ((裏当金はない))
        ((両側探傷は10dBより大きい))
        ((欠陥位置は表面付近である))
        ((反射点は裏波付近である)))
    (then ((欠陥ではなく妨害エコー2である))) )
(rule 妨害エコー3
  (if ((開先形状はV形開先))
        ((裏当金はある))
        ((両側探傷は10dBより大きい))
        ((欠陥位置は裏当金付近である))
        ((反射点はルート部である)))
    (then ((欠陥ではなく妨害エコー3である))) )
(rule 妨害エコー4
  (if ((開先形状はV形開先))
        ((裏当金はある))
        ((両側探傷は10dBより大きい))
        ((欠陥位置は裏当金付近である))
        ((反射点は裏当金溶接付近である)))
    (then ((欠陥ではなく妨害エコー4である))) )
(rule 妨害エコー5
  (if ((開先形状はV形開先))
        ((裏当金はある))
        ((両側探傷は10dBより大きい))
        ((欠陥位置は裏当金溶接付近である)))
    (then ((欠陥ではなく妨害エコー5である))) )
(rule 妨害エコー6
  (if ((開先形状はV形開先))
        ((裏当金はある))
        ((両側探傷は10dBより大きい))
        ((欠陥位置は裏当金溶接裏波付近である)))
    (then ((欠陥ではなく妨害エコー6である))) )
(rule 妨害エコー7
  (if ((開先形状はレ形開先))
        ((両側探傷は10dBより大きい))
        ((欠陥位置は裏波付近である)))
    (then ((欠陥ではなく妨害エコー1である))) )
(rule 妨害エコー8
  (if ((開先形状はレ形開先))
        ((裏当金はない))
        ((両側探傷は10dBより大きい))
        ((欠陥位置は表面付近である))
        ((反射点は裏波付近である)))
    (then ((欠陥ではなく妨害エコー2である))) )
(rule 妨害エコー9
  (if ((開先形状はレ形開先))
        ((裏当金はある))
        ((両側探傷は10dBより大きい))
        ((欠陥位置は裏当金付近である))
        ((反射点はルート部である)))
    (then ((欠陥ではなく妨害エコー3である))) )
(rule 妨害エコー10
  (if ((開先形状はレ形開先))
        ((裏当金はある))
        ((両側探傷は10dBより大きい))
        ((欠陥位置は裏当金溶接付近である)))
    (then ((欠陥ではなく妨害エコー4である))) )
(rule 妨害エコー11
  (if ((開先形状はレ形開先))
        ((裏当金はある))
        ((両側探傷は10dBより大きい))

```

```

      ((欠陥位置は裏当金溶接付近である)))
    (then ((欠陥ではなく妨害エコー5である))) )
(rule 妨害エコー12
  (if ((開先形状はレ形開先))
        ((裏当金はある))
        ((両側探傷は10dBより大きい))
        ((欠陥位置は裏当金溶接裏波付近である)))
    (then ((欠陥ではなく妨害エコー6である))) )
(rule 妨害エコー13
  (if ((継手形状はT字継手))
        ((開先形状はレ形開先))
        ((欠陥位置は開先裏波付近である)))
    (then ((欠陥ではなく妨害エコー7である))) )
(rule 妨害エコー14
  (if ((継手形状はT字継手))
        ((開先形状はレ形開先))
        ((裏当金はある))
        ((路程長さは1回反射である))
        ((欠陥位置は開先付近である))
        ((反射点は裏当金溶接付近である)))
    (then ((欠陥ではなく妨害エコー8である))) )

```

```

))))

```

```

(rds b:ndi_e.kdb)

```



```
**** mulisp86 version ***** 92.11.14 *****
:
: File name : NDI E. KDB
: Feature : 欠陥の等級判別および欠陥の寸法の計算に関するルール
:
: ****
```

```
(store_frames '(
  (figure_rules ;欠陥の形状に関するプロダクションルールを格納するフレーム
    (ako (value trigrules))
    (forward (value ut_rules))
    (input (value *result_ut*))
    (result (value *result_ef*))
    (whylog (value *whylog_ef*))
    (rules (value
      (rule 欠陥形状ルート割れ
        (if ((欠陥はルート割れである)))
        (then ((欠陥形状は帯状平面欠陥))))
      (rule 欠陥形状ルート溶け込み不良
        (if ((欠陥はルート溶け込み不良である)))
        (then ((欠陥形状は帯状平面欠陥))))
      (rule 欠陥形状密集ブローホール
        (if ((欠陥は密集ブローホールである)))
        (then ((欠陥形状は球状欠陥))))
      (rule 欠陥形状単独ブローホール
        (if ((欠陥は単独ブローホールである)))
        (then ((欠陥形状は球状欠陥))))
      (rule 欠陥形状開先面スラグ巻き込み
        (if ((欠陥は開先面スラグ巻き込みである)))
        (then ((欠陥形状は帯状平面欠陥))))
      (rule 欠陥形状層間スラグ巻き込み
        (if ((欠陥は層間スラグ巻き込みである)))
        (then ((欠陥形状は帯状平面欠陥))))
      (rule 欠陥形状スラグ巻き込み
        (if ((欠陥はスラグ巻き込みである)))
        (then ((欠陥形状は帯状平面欠陥))))
      (rule 欠陥形状内部溶け込み不良
        (if ((欠陥は内部溶け込み不良である)))
        (then ((欠陥形状は帯状平面欠陥))))
      (rule 欠陥形状内部割れ
        (if ((欠陥は内部割れである)))
        (then ((欠陥形状は帯状平面欠陥))))
      (rule 欠陥形状開先面融合不良
        (if ((欠陥は開先面融合不良である)))
        (then ((欠陥形状は帯状平面欠陥))))
      (rule 欠陥形状層間融合不良
        (if ((欠陥は層間融合不良である)))
        (then ((欠陥形状は帯状平面欠陥))))
      (rule 欠陥形状止端割れ
        (if ((欠陥は止端割れである)))
        (then ((欠陥形状は帯状平面欠陥))))
      (rule 欠陥形状アンダーカット
        (if ((欠陥はアンダーカットである)))
        (then ((欠陥形状は直線溝欠陥))))
      (rule 欠陥形状ピット
        (if ((欠陥はピットである)))
        (then ((欠陥形状は縦穴欠陥))))
      (rule 欠陥形状ビード表面割れ
        (if ((欠陥はビード表面割れである)))
        (then ((欠陥形状は帯状平面欠陥))))
      (rule 欠陥形状ビード下割れ
        (if ((欠陥はビード下割れである)))
        (then ((欠陥形状は帯状平面欠陥))))))
    )
  )
```

```
(flaw_class ;欠陥の等級分類に関するプロダクションルールを格納するフレーム
  (ako (value trigrules2))
  (forward (value *result_ut2*))
  (input (value *result_ut2*))
  (result (value *result_ec*))
  (whylog (value *whylog_ec*))
  (rules (value
    (rule 等級1
      (if ((E区分はM線以下))
        ((板厚Tは18以下である))
        ((欠陥指示長さは4以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級2
      (if ((E区分はM線以下))
        ((板厚Tは18以下である))
        ((欠陥指示長さは6以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級3
      (if ((E区分はM線以下))
        ((板厚Tは18を越え60未満である))
        ((欠陥指示長さはT/4以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級4
```

```

      (if ((E区分はM線以下))
        ((板厚Tは18を越え60未満である))
        ((欠陥指示長さはT/3以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級5
      (if ((E区分はM線以下))
        ((板厚Tは60以上である))
        ((欠陥指示長さは15以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級6
      (if ((E区分はM線以下))
        ((板厚Tは60以上である))
        ((欠陥指示長さは20以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級7
      (if ((E区分はM線を越えH線以下))
        ((板厚Tは18以下である))
        ((欠陥指示長さは4以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級8
      (if ((E区分はM線を越えH線以下))
        ((板厚Tは18以下である))
        ((欠陥指示長さは6以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級9
      (if ((E区分はM線を越えH線以下))
        ((板厚Tは18を越え60未満である))
        ((欠陥指示長さはT/4以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級10
      (if ((E区分はM線を越えH線以下))
        ((板厚Tは18を越え60未満である))
        ((欠陥指示長さはT/3以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級11
      (if ((E区分はM線を越えH線以下))
        ((板厚Tは60以上である))
        ((欠陥指示長さは15以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級12
      (if ((E区分はM線を越えH線以下))
        ((板厚Tは60以上である))
        ((欠陥指示長さは20以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級13
      (if ((E区分はH線を越える))
        ((板厚Tは18以下である))
        ((欠陥指示長さは4以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級14
      (if ((E区分はH線を越える))
        ((板厚Tは18を越え60未満である))
        ((欠陥指示長さはT/4以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級15
      (if ((E区分はH線を越える))
        ((板厚Tは60以上である))
        ((欠陥指示長さは15以下である)))
      (then ((欠陥の等級は1級である))))
    (rule 等級16
      (if ((E区分はM線以下))
        ((板厚Tは18以下である))
        ((欠陥指示長さは9以下である)))
      (then ((欠陥の等級は2級である))))
    (rule 等級17
      (if ((E区分はM線以下))
        ((板厚Tは18を越え60未満である))
        ((欠陥指示長さはT/2以下である)))
      (then ((欠陥の等級は2級である))))
    (rule 等級18
      (if ((E区分はM線を越えH線以下))
        ((板厚Tは60以上である))
        ((欠陥指示長さは30以下である)))
      (then ((欠陥の等級は2級である))))
    (rule 等級19
      (if ((E区分はM線を越えH線以下))
        ((板厚Tは18以下である))
        ((欠陥指示長さは9以下である)))
      (then ((欠陥の等級は2級である))))
    (rule 等級20
      (if ((E区分はM線を越えH線以下))
        ((板厚Tは18を越え60未満である))
        ((欠陥指示長さはT/2以下である)))
      (then ((欠陥の等級は2級である))))
    (rule 等級21
      (if ((E区分はM線を越えH線以下))
        ((板厚Tは60以上である))
        ((欠陥指示長さは30以下である)))
      (then ((欠陥の等級は2級である))))
  )
)
```







\*\*\*\* mulisp86 version \*\*\*\*\* 92.11.14 \*\*\*\*\*

File name : NDI\_FUZZY.KDB  
Feature : メンバシップ関数の定義

\*\*\*\*\*

(store\_frames

(membership ;メンバーシップ関数を定義するフレーム

(name (value 両側探傷

点数

振子走査

エコー

波形

特性曲線)))

(両側探傷

(ako (value membership))

(name (value 10dB以下である

10dBより大きい

10dBより明らかに小さい

10dBより明らかに大きい

10dB前後である

測定できないまたは測定していない)))

(点数

(ako (value membership))

(name (value 約6以下である

約7以上約9以下である

約10以上である

約4である

約5である

約6である

約7である

約8である

約9である

約10である

約11である

約12である)))

(振子走査

(ako (value membership))

(name (value 小さい

8dB未満

8dB以上15dB以下

15dBを超える)))

(エコー

(ako (value membership))

(name (value いろいろなビーム路程で立ち上がる

ギザギザである

ギザギザでない

ギザギザに近い

ギザギザでないに近い

いろいろなビーム路程で立ち上がらない

明らかにいろいろなビーム路程で立ち上がる

いろいろなビーム路程で立ち上がるかギザギザであるかはつきりしない

明らかにギザギザであるかはつきりしない

ギザギザかどうかはつきりしない

明らかにギザギザでない)))

(特性曲線

(ako (value membership))

(name (value A2-φ4より緩やかである

A2-φ4より緩やかでない

明らかにA2-φ4より緩やかである

A2-φ4とほぼ同じ傾きである

A2-φ4より緩やかかどうかはつきりしない

測定していない)))

(10dB以下である

(ako (value 両側探傷))

(type (value 左台形))

(屈折点 (value 10.0))

(幅 (value 2.0)))

(10dBより大きい

(ako (value 両側探傷))

(type (value 右台形))

(屈折点 (value 10.0))

(幅 (value 2.0)))

(10dBより明らかに小さい

(ako (value 両側探傷))

(type (value 左台形))

(屈折点 (value 7.0))

(幅 (value 2.0)))

(10dBより明らかに大きい

(ako (value 両側探傷))

(type (value 右台形))

(屈折点 (value 13.0))

(幅 (value 2.0)))

(10dB前後である

(ako (value 両側探傷))

(type (value 三角形))

(中心点 (value 10.0))

(幅 (value 2.0)))

(測定できないまたは測定していない

(ako (value 両側探傷))

(type (value 三角形))

(中心点 (value 10.0))

(幅 (value 2.0)))

(約6以下である

(ako (value 点数))

(type (value 左台形))

(屈折点 (value 6.0))

(幅 (value 2.0)))

(約7以上約9以下である

(ako (value 点数))

(type (value 台形))

(左屈折点 (value 7.0))

(右屈折点 (value 9.0))

(左幅 (value 4.0))

(右幅 (value 4.0)))

(約10以上である

(ako (value 点数))

(type (value 右台形))

(屈折点 (value 10.0))

(幅 (value 2.0)))

(約4である

(ako (value 点数))

(type (value 三角形))

(中心点 (value 4.0))

(幅 (value 2.0)))

(約5である

(ako (value 点数))

(type (value 三角形))

(中心点 (value 5.0))

(幅 (value 2.0)))

(約6である

(ako (value 点数))

(type (value 三角形))

(中心点 (value 6.0))

(幅 (value 2.0)))

(約7である

(ako (value 点数))

(type (value 三角形))

(中心点 (value 7.0))

(幅 (value 2.0)))

(約8である

(ako (value 点数))

(type (value 三角形))

(中心点 (value 8.0))

(幅 (value 2.0)))

(約9である

(ako (value 点数))

(type (value 三角形))

(中心点 (value 9.0))

(幅 (value 2.0)))

(約10である

(ako (value 点数))

(type (value 三角形))

(中心点 (value 10.0))

(幅 (value 2.0)))

(約11である

(ako (value 点数))

(type (value 三角形))

(中心点 (value 11.0))

(幅 (value 2.0)))

(約12である

(ako (value 点数))

(type (value 三角形))

(中心点 (value 12.0))



```

(幅 (value 2.0)))

(小さい
(ako (value 振子走査))
(type (value 左台形))
(屈折点 (value 5.0))
(幅 (value 3.0)))

(8 d B未満
(ako (value 振子走査))
(type (value 左台形))
(屈折点 (value 8.0))
(幅 (value 2.0)))

(8 d B以上15 d B以下
(ako (value 振子走査))
(type (value 台形))
(左屈折点 (value 8.0))
(右屈折点 (value 15.0))
(左幅 (value 2.0))
(右幅 (value 2.0)))

(15 d Bを越える
(ako (value 振子走査))
(type (value 右台形))
(屈折点 (value 15.0))
(幅 (value 2.0)))

(いろいろなビーム路程で立ち上がる
(ako (value エコー))
(type (value 右台形))
(屈折点 (value 5.6))
(幅 (value 5.2)))

(ギザギザである
(ako (value エコー))
(type (value 台形))
(左屈折点 (value -1.0))
(右屈折点 (value 3.0))
(左幅 (value 2.0))
(右幅 (value 3.0)))

(ギザギザでない
(ako (value エコー))
(type (value 左台形))
(屈折点 (value -3.0))
(幅 (value 2.0)))

(ギザギザに近い
(ako (value エコー))
(type (value 台形))
(左屈折点 (value 0.5))
(右屈折点 (value 4.0))
(左幅 (value 6.0))
(右幅 (value 2.0)))

(ギザギザでないに近い
(ako (value エコー))
(type (value 左台形))
(屈折点 (value 2.8))
(幅 (value 2.0)))

(いろいろなビーム路程で立ち上がらない
(ako (value エコー))
(type (value 左台形))
(屈折点 (value 4.0))
(幅 (value 2.0)))

(明らかにいろいろなビーム路程で立ち上がる
(ako (value エコー))
(type (value 右台形))
(屈折点 (value 6.0))
(幅 (value 2.0)))

(いろいろなビーム路程で立ち上がるかギザギザであるかはつきりしない
(ako (value エコー))
(type (value 三角形))
(中心点 (value 3.0))
(幅 (value 2.0)))

(明らかにギザギザである
(ako (value エコー))
(type (value 三角形))
(中心点 (value 0.0))
(幅 (value 2.0)))

(ギザギザかどうかはつきりしない
(ako (value エコー))

```

```

(type (value 三角形))
(中心点 (value -1.0))
(幅 (value 3.0)))

(明らかにギザギザでない
(ako (value エコー))
(type (value 左台形))
(屈折点 (value -2.0))
(幅 (value 4.0)))

(A2-φ4より緩やかである
(ako (value 特性曲線))
(type (value 左台形))
(屈折点 (value 0.0))
(幅 (value 2.0)))

(A2-φ4より緩やかでない
(ako (value 特性曲線))
(type (value 右台形))
(屈折点 (value 0.0))
(幅 (value 2.0)))

(明らかにA2-φ4より緩やかである
(ako (value 特性曲線))
(type (value 左台形))
(屈折点 (value -3.0))
(幅 (value 2.0)))

(A2-φ4とほぼ同じ傾きかもしくは急である
(ako (value 特性曲線))
(type (value 右台形))
(屈折点 (value 3.0))
(幅 (value 2.0)))

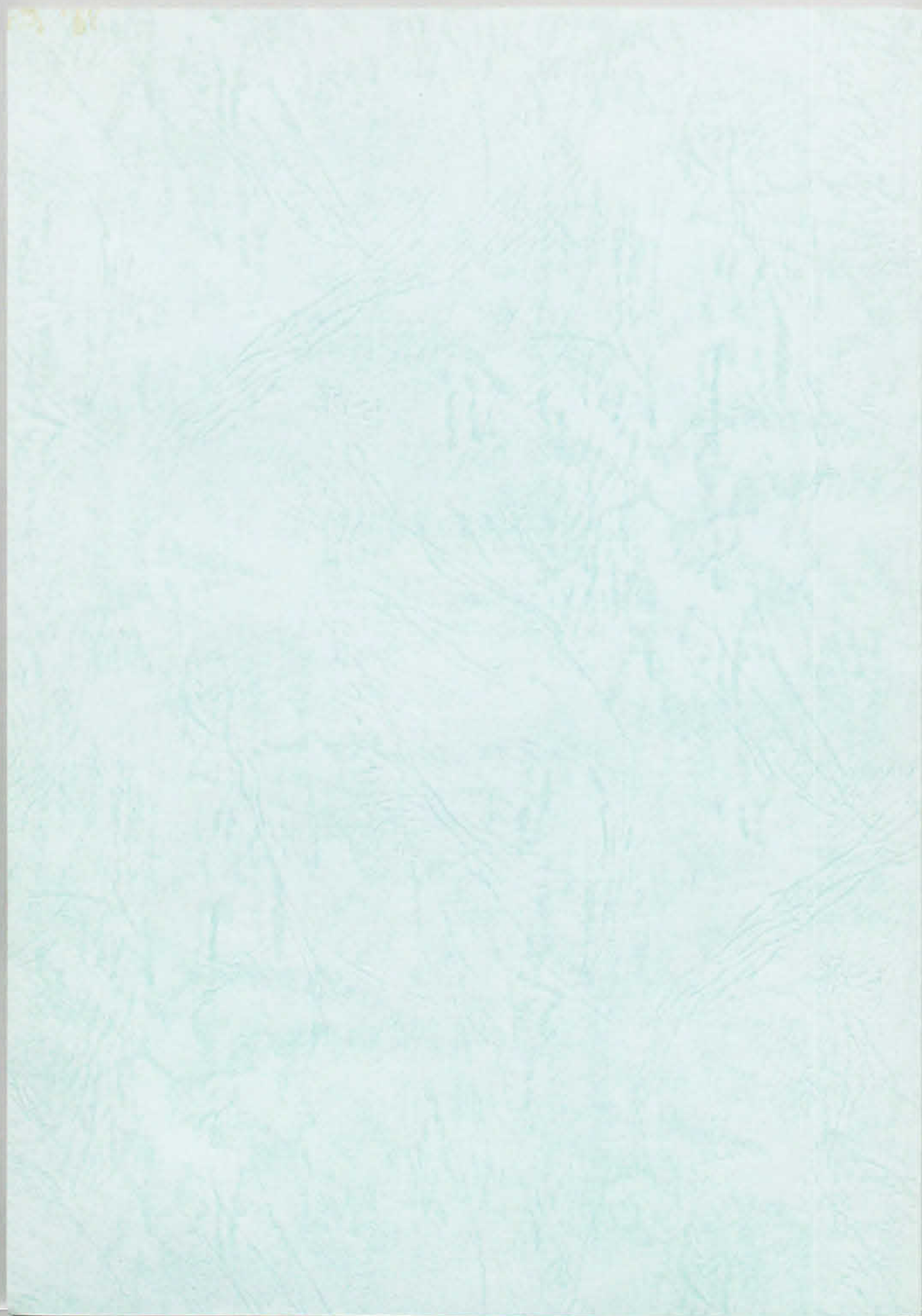
(A2-φ4より緩やかかどうかはつきりしない
(ako (value 特性曲線))
(type (value 三角形))
(中心点 (value 0.0))
(幅 (value 2.0)))

(測定していない
(ako (value 特性曲線))
(type (value 三角形))
(中心点 (value 0.0))
(幅 (value 2.0)))

```

(rds)







様式9

## 論文審査の結果の要旨

報告番号	甲 工 乙 工 第 25 号 氏 名 松 浦 洋 司
審査委員	主 査 芳 村 敏 夫 教授 副 査 三 輪 恵 教授 副 査 宮 本 定 明 教授
学位論文題目	超音波探傷試験評価のためのファジイエキスパートシステムの構築に関する研究
審査結果の要旨	<p>各種溶接構造物の溶接欠陥の検査として超音波探傷試験は作業性の良さや装置の性能の向上などにより最近広く用いられるようになってきている。この試験は、試験結果から欠陥の種類、欠陥寸法および欠陥等級を推定し、欠陥の合否を判定するものであるが、推定規程にない点が多くなり、検査技術者の主観の差による推定結果の差異が問題となる。このため、超音波探傷試験評価のためのファジイエキスパートシステムの構築に関する研究を行っている。</p> <p>まず、このシステムはワークステーション上に、Common Lisp 言語を用いて開発し、推論エンジン、知識ベースおよびワーキングメモリから構成されている。ここでは、黑板モデルを採用し、知識はIF-THEN形式のプロダクションルールで表現し、フレームモデルによってグルーピング化している。推論にはあいまいさを扱うことができ、ファジイ推論を用いており、開発したシステムの推論結果と検査技術者の推定結果との比較から、推論精度の高さが確認されている。</p> <p>次に、ファジイ推論の中で問題となる結合演算子の選定およびメンバシップ関数の改良について考察している。結合演算子は本システムに適用したとき最も推論精度の良いものを選定し、メンバシップ関数は複数の検査技術者による知識を調整して作成している。これによって、先に開発されたシステムにみられた欠陥が解消され、精度が一段と向上している。</p> <p>最後に、持ち運びが可能なパソコン用のシステムの構築および妨害エコーについての知識の補充を行っている。妨害エコー判定用の知識に含まれるあいまいさについてはメンバシップ関数等を用いて扱い、検査技術者の推定結果との比較により、妨害エコーについても高精度で推論可能となっている。</p> <p>以上本研究は、超音波探傷試験の結果から欠陥の状況を推論できるファジイエキスパートシステムを構築するものであり、その推定精度の高さや有用性から本論文は博士（工学）の学位授与に値するものと判定する。</p>